

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

**RESTAURACE SIGNÁLU S OMEZENOU OKAMŽITOU  
HODNOTOU PRO VÍCEKANÁLOVÝ AUDIO SIGNÁL**

RESTORATION OF SIGNALS WITH LIMITED INSTANTANEOUS VALUE FOR THE MULTICHANNEL AUDIO  
SIGNAL

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Vojtěch Hájek**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Pavel Záviška**

**BRNO 2019**

# Diplomová práce

magisterský navazující studijní obor **Audio inženýrství**

Ústav telekomunikací

**Student:** Bc. Vojtěch Hájek

**ID:** 174302

**Ročník:** 2

**Akademický rok:** 2018/19

## NÁZEV TÉMATU:

### **Restaurace signálu s omezenou okamžitou hodnotou pro vícekanálový audio signál**

#### POKYNY PRO VYPRACOVÁNÍ:

Prostudujte znehodnocení zvukového signálu s omezenou okamžitou hodnotou, tedy jev nazývaný též jako clipping. Problém analyzujte a seznamte se s obvyklými metodami rekonstrukce takovýchto signálů. Prostudujte především metody založené na tzv. řídkých reprezentacích signálů.

Následně v prostředí MATLAB implementujte rekonstrukční algoritmus pro "declipping" na bázi řídkých reprezentací signálů. Algoritmus nejprve použijte pro monofonní audio signál a následně jej vhodně rozšiřte s využitím strukturované řídkosti pro použití na vícekanálový audio signál. Porovnejte analyzující a syntetizující přístup k rekonstrukci a proveďte důkladnou analýzu nastavení různých parametrů algoritmu za účelem dosažení co nejlepší kvality rekonstrukce. Výsledky rekonstrukce pak vyhodnoťte pomocí vhodných objektivních i subjektivních ukazatelů. Srovnajte také vícekanálový přístup a jednokanálový přístup aplikovaný na každý kanál zvlášť.

#### DOPORUČENÁ LITERATURA:

[1] Hrbáček, R., Rajmic, P., Veselý, V., Špiřík, J. Řídké reprezentace signálů: úvod do problematiky, Elektrevue, 2011. ISSN 1213-1539.

[2] Gaultier, C., Bertin, N., Gribonval, R. CASCADE : Channel-Aware Structured Cospase Audio DEclipper. ICASSP 2018 - IEEE International Conference on Acoustics, Speech and Signal Processing, Apr 2018, Calgary, Canada. pp.1-5, 2018.

**Termín zadání:** 1.2.2019

**Termín odevzdání:** 16.5.2019

**Vedoucí práce:** Ing. Pavel Závíška

**Konzultant:**

**prof. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

#### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Tato diplomová práce se zabývá rekonstrukcí saturovaného vícekanálového audio signálu pomocí metod založených na řídké reprezentaci signálu. V první části práce je popsána teorie clippingu u audio signálů a teorie řídké reprezentace signálů. V této části je obsažena také krátká rešerše současných rekonstrukčních algoritmů. Následně jsou představeny dva rekonstrukční algoritmy, které byly v rámci práce naprogramovány v prostředí Matlab. První z nich je algoritmus SPADE, „state-of-the-art“ pro rekonstrukci monofonních signálů, a druhým je od něj odvozený algoritmus CASCADE, navržený pro vícekanálové signály. Ve třetí části práce jsou oba algoritmy otestovány a porovnány pomocí objektivních ukazatelů SDR a PEAQ a pomocí subjektivního poslechového testu MUSHRA.

## KLÍČOVÁ SLOVA

clipping, declipping, zpracování signálu, řídká reprezentace signálu, SPADE, CASCADE

## ABSTRACT

This master's thesis deals with the restoration of clipped multichannel audio signals based on sparse representations. First, a general theory of clipping and theory of sparse representations of audio signals is described. A short overview of existing restoration methods is part of this thesis as well. Subsequently, two declipping algorithms are introduced and are also implemented in the Matlab environment as a part of the thesis. The first one, SPADE, is considered a state-of-the-art method for mono audio signals declipping and the second one, CASCADE, which is derived from SPADE, is designed for the restoration of multichannel signals. In the last part of the thesis, both algorithms are tested and the results are compared using the objective measures SDR and PEAQ, and also using the subjective listening test MUSHRA.

## KEYWORDS

clipping, declipping, signal processing, sparse representation, SPADE, CASCADE

HÁJEK, Vojtěch. *Restaurace signálu s omezenou okamžitou hodnotou pro vícekanálový audio signál*. Brno, Rok, 77 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Pavel Záviška

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Restaurace signálu s omezenou okamžitou hodnotou pro vícekanálový audio signál“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Pavlu Záviškovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....

podpis autora

# Obsah

<b>Úvod</b>	<b>12</b>
<b>1 Clipping</b>	<b>13</b>
1.1 Hard clipping . . . . .	13
1.2 Soft clipping . . . . .	14
1.3 Důsledky clippingu . . . . .	15
1.3.1 Harmonické zkreslení . . . . .	15
1.3.2 Aliasing . . . . .	16
1.3.3 Intermodulační zkreslení . . . . .	16
1.4 Příklady clippingu v praxi . . . . .	16
<b>2 Declipping</b>	<b>18</b>
2.1 Přehled existujících algoritmů . . . . .	18
2.1.1 Algoritmy využívající statistický model . . . . .	18
2.1.2 Algoritmy využívající interpolaci . . . . .	18
2.1.3 Algoritmy využívající řídkou reprezentaci signálu . . . . .	18
2.1.4 Algoritmy využívající jiné metody . . . . .	19
<b>3 Řídká reprezentace signálu</b>	<b>21</b>
3.1 Základní pojmy . . . . .	21
3.2 Báze a framy . . . . .	22
3.2.1 Vektorový prostor . . . . .	22
3.2.2 Báze vektorového prostoru . . . . .	22
3.2.3 Framy . . . . .	22
3.3 Řídká reprezentace audio signálu . . . . .	23
3.4 Strukturovaná řídkost . . . . .	23
3.4.1 Strukturovaná řídkost v audio signálech . . . . .	24
3.5 Hledání řídkého řešení . . . . .	24
3.5.1 Hladové algoritmy . . . . .	24
3.5.2 Relaxační algoritmy . . . . .	25
3.5.3 Hybridní algoritmy . . . . .	25
<b>4 Hledání řídkého řešení v praxi</b>	<b>27</b>
4.1 Segmentace signálu . . . . .	27
4.2 Rozdělení vzorků saturovaného signálu . . . . .	27
4.3 Algoritmus SPADE . . . . .	29
4.4 Algoritmus CASCADE . . . . .	31



<b>5</b>	<b>Algoritmus</b>	<b>33</b>
5.1	Declipping.m . . . . .	33
5.2	Segmentation.m . . . . .	34
5.3	Rekonstrukční algoritmy . . . . .	34
5.4	Pomocné funkce . . . . .	35
<b>6</b>	<b>Hodnocení výsledků algoritmu</b>	<b>37</b>
6.1	Testovací nahrávky . . . . .	37
6.2	Objektivní kritéria hodnocení . . . . .	37
6.2.1	Signal to Distortion Ratio . . . . .	37
6.2.2	Perceptual Evaluation of Audio Quality . . . . .	38
6.3	Subjektivní kritéria hodnocení . . . . .	39
6.3.1	MUSHRA . . . . .	39
<b>7</b>	<b>Výsledky</b>	<b>41</b>
7.1	Testy algoritmu CASCADE . . . . .	41
7.1.1	Nastavení délky okna . . . . .	41
7.1.2	Nastavení posuvu okna . . . . .	42
7.1.3	Nastavení redundance . . . . .	43
7.1.4	Nastavení parametru $\alpha$ . . . . .	45
7.1.5	Nastavení parametru $\beta$ . . . . .	46
7.1.6	Rekonstrukce korelovaných kanálů signálu zvlášť . . . . .	47
7.2	Nastavení parametrů algoritmu SPADE . . . . .	49
7.3	Porovnání algoritmů SPADE a CASCADE . . . . .	50
7.3.1	Výsledky SDR . . . . .	50
7.3.2	Výsledky PEAQ . . . . .	51
7.3.3	Výsledky MUSHRA . . . . .	53
<b>8</b>	<b>Závěr</b>	<b>55</b>
	<b>Literatura</b>	<b>57</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>60</b>
	<b>Seznam příloh</b>	<b>61</b>
<b>A</b>	<b>Podrobné hodnocení výsledků rekonstrukčních algoritmů</b>	<b>62</b>
A.1	Ukazatel SDR . . . . .	62
A.2	Metoda PEAQ . . . . .	68
A.3	Subjektivní poslechový test . . . . .	74
<b>B</b>	<b>Obsah příloženého CD</b>	<b>77</b>

## Seznam obrázků

1.1	Převodní charakteristika clippingu. . . . .	13
1.2	Hard clipping – průběh signálu a porovnání spekter. . . . .	14
1.3	Soft clipping – průběh signálu a porovnání spekter. . . . .	15
1.4	Intermodulační zkreslení – průběh signálu a porovnání spekter. . . . .	16
3.1	Spektrogram sociálně řídkého signálu. . . . .	24
4.1	Hannova okna s překryvem 50 %. . . . .	28
4.2	Hannova okna s překryvem 75 %. . . . .	28
6.1	Blokové schéma základní verze percepčního modelu PEAQ. . . . .	38
6.2	Blokové schéma rozšířené verze percepčního modelu PEAQ. . . . .	39
7.1	Závislost $\Delta\text{SDR}$ na délce okna – A-CASCADE . . . . .	42
7.2	Závislost $\Delta\text{SDR}$ na posuvu okna – A-CASCADE . . . . .	43
7.3	Závislost rychlosti výpočtu na posuvu okna – A-CASCADE . . . . .	43
7.4	Závislost $\Delta\text{SDR}$ na redundanci transformace – CASCADE . . . . .	44
7.5	Závislost rychlosti výpočtu na redundanci transformace – CASCADE . . . . .	45
7.6	Průběh progresivního parametru $\alpha$ . . . . .	46
7.7	Závislost $\Delta\text{SDR}$ na hodnotě $\alpha$ – A-CASCADE . . . . .	46
7.8	Závislost $\Delta\text{SDR}$ na hodnotě $\beta$ – A-CASCADE . . . . .	47
7.9	Závislost $\Delta\text{SDR}$ LFE kanálu na rekonstrukčním algoritmu . . . . .	48
7.10	Hodnota $\Delta\text{SDR}$ jednotlivých kanálů signálu S6. . . . .	49
7.11	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu . . . . .	51
7.12	Závislost ODG na rekonstrukčním algoritmu . . . . .	52
7.13	Závislost ODG LFE kanálu na rekonstrukčním algoritmu . . . . .	53
7.14	Průměrné hodnocení kvality nahrávek podle poslechového testu MUSHRA. . . . .	54
A.1	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S1 . . . . .	62
A.2	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S2 . . . . .	62
A.3	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S3 . . . . .	63
A.4	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S4 (část 1) . . . . .	63
A.5	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S4 (část 2) . . . . .	64
A.6	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S4 (část 3) . . . . .	64
A.7	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S5 (část 1) . . . . .	65
A.8	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S5 (část 2) . . . . .	65
A.9	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S5 (část 3) . . . . .	66
A.10	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S6 (část 1) . . . . .	66
A.11	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S6 (část 2) . . . . .	67
A.12	Závislost $\Delta\text{SDR}$ na rekonstrukčním algoritmu - S6 (část 3) . . . . .	67
A.13	Závislost ODG na rekonstrukčním algoritmu - S1 . . . . .	68
A.14	Závislost ODG na rekonstrukčním algoritmu - S2 . . . . .	68
A.15	Závislost ODG na rekonstrukčním algoritmu - S3 . . . . .	69
A.16	Závislost ODG na rekonstrukčním algoritmu - S4 (část 1) . . . . .	69
A.17	Závislost ODG na rekonstrukčním algoritmu - S4 (část 2) . . . . .	70

A.18 Závislost ODG na rekonstrukčním algoritmu - S4 (část 3) . . . . .	70
A.19 Závislost ODG na rekonstrukčním algoritmu - S5 (část 1) . . . . .	71
A.20 Závislost ODG na rekonstrukčním algoritmu - S5 (část 2) . . . . .	71
A.21 Závislost ODG na rekonstrukčním algoritmu - S5 (část 3) . . . . .	72
A.22 Závislost ODG na rekonstrukčním algoritmu - S6 (část 1) . . . . .	72
A.23 Závislost ODG na rekonstrukčním algoritmu - S6 (část 2) . . . . .	73
A.24 Závislost ODG na rekonstrukčním algoritmu - S6 (část 3) . . . . .	73

## Seznam tabulek

6.1	Hodnocení kvality signálu podle stupnice ODG. . . . .	38
6.2	Hodnocení kvality signálu podle stupnice MUSHRA. . . . .	40
7.1	Výchozí nastavení parametrů rekonstrukce pomocí algoritmu CASCADE . .	41
7.2	Nastavení algoritmu SPADE pro rekonstrukci nízkofrekvenčního kanálu. . .	48
7.3	Nastavení algoritmů SPADE a CASCADE pro srovnávací test. . . . .	50
7.4	Porovnání efektivity algoritmů SPADE a CASCADE. . . . .	51
A.1	Výsledky subjektivního poslechového testu - část 1 ze 3. . . . .	74
A.2	Výsledky subjektivního poslechového testu - část 2 ze 3. . . . .	75
A.3	Výsledky subjektivního poslechového testu - část 3 ze 3. . . . .	76

# Úvod

Tzv. clipping je jedním z nejběžnějších druhů poškození audio signálu. Může vzniknout jak v analogové, tak v digitální doméně hned z několika důvodů: například přebuzení zesilovače nebo malý dynamický rozsah některých prvků zvukového řetězce. Důsledkem clippingu je nelineární zkreslení, tedy vznik vyšších harmonických kmitočtů ve spektru signálu. Teoretickému popisu clippingu a jeho důsledkům se věnuje první kapitola.

Rekonstrukce zvukového záznamu poškozeného clippingem je poměrně obtížná. Využívají se k ní moderní matematické metody. První metody, zabývající se tzv. declippingem pochází z 80. let minulého století. V současnosti se k declippingu nejčastěji využívají algoritmy založené na tzv. řídké reprezentaci signálů. Tyto metody jsou shrnuty v krátké rešerši ve druhé kapitole.

Třetí kapitola obsahuje teoretický základ pro pochopení problematiky řídké reprezentace signálu a hledání řídkého řešení. Podstatnou částí této kapitoly je část zabývající se řešením optimalizační úlohy metodou ADMM.

Ve čtvrté kapitole je teoreticky popsáno hledání řídkého řešení za pomoci algoritmu SPADE, který je navržen pro rekonstrukci monofonních signálů a z něj odvozeného algoritmu CASCADE, který je v podstatě jeho rozšířením pro vícekanálové signály. Implementace těchto algoritmů včetně všech pomocných funkcí v programu Matlab je popsána v následující kapitole.

Šestá kapitola se věnuje problematice hodnocení kvality signálu. Jsou zde popsány testovací signály, na kterých byla ověřena funkčnost obou rekonstrukčních algoritmů a také metody, kterými byla kvalita rekonstruovaného signálu hodnocena.

V poslední, sedmé, kapitole je popsán proces hledání nejefektivnějšího nastavení algoritmu CASCADE i SPADE a jsou zde prezentovány průběžné výsledky. Následně jsou zde oba algoritmy porovnány jak pomocí objektivních měřítek, tak pomocí subjektivního poslechového testu.

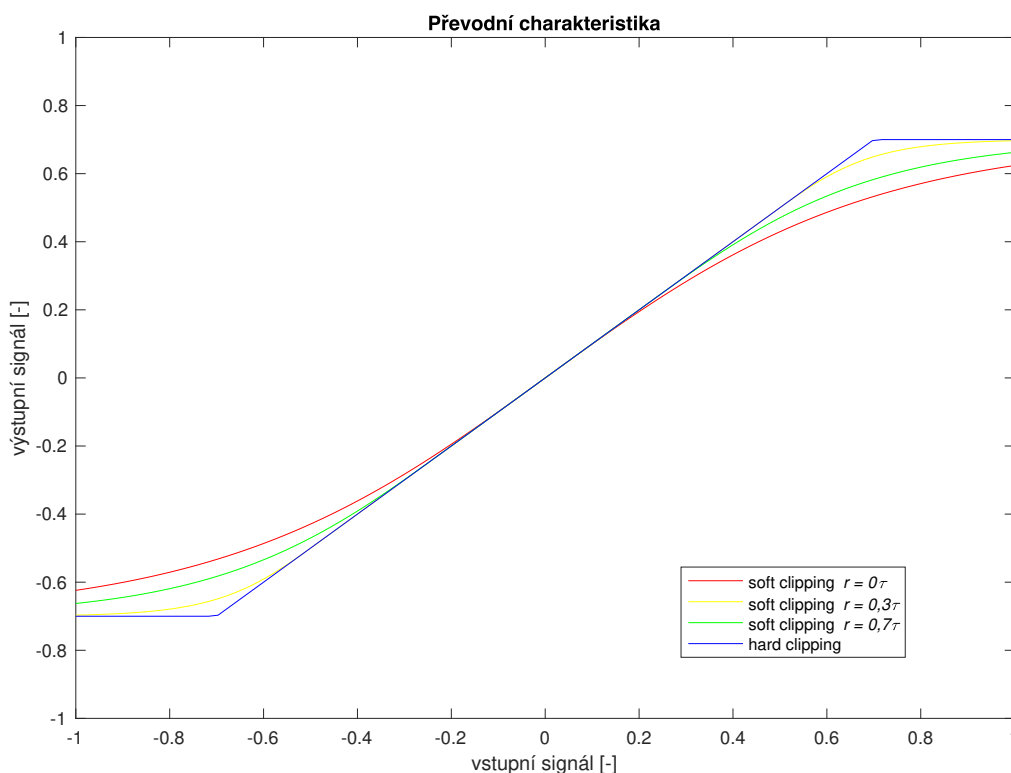
Pro větší přehlednost jsou v této kapitole prezentovány pouze průměrné výsledky pro všechny signály. Podrobné výsledky jsou součástí přílohy této práce.

# 1 Clipping

V této kapitole se budu zabývat problematikou clippingu v oblasti zpracování audio signálu. Podrobněji rozeberu dva základní typy clippingu: hard clipping a soft clipping. Popíši zde, kde nejčastěji ke clippingu dochází a jaké jsou jeho následky. V této kapitole čerpám především z [1], [2] a [3].

Clipping je druh poškození signálu, při kterém dochází k oříznutí kladných i záporných špiček signálu, které v absolutní hodnotě úrovně přesahují výchylku  $\tau$ . Ke clippingu může dojít v aplikacích, ve kterých během zpracování signálu dochází k jeho zesílení nebo během A/D převodu signálu. Pravděpodobně jednou z nejčastějších příčin vzniku clippingu je špatné nastavení vstupní úrovně signálu během nahrávání zvukového záznamu.

Podle tvaru převodní charakteristiky systému rozlišujeme dva druhy clippingu: hard clipping a soft clipping. Hard clipping se vyznačuje tvrdým oříznutím špiček nad úrovní  $\tau$ , zatímco signál pod touto úrovní zůstává neovlivněn. Při soft clippingu dochází ke kompresi signálu již pod touto úrovní.



Obr. 1.1: Převodní charakteristika clippingu.

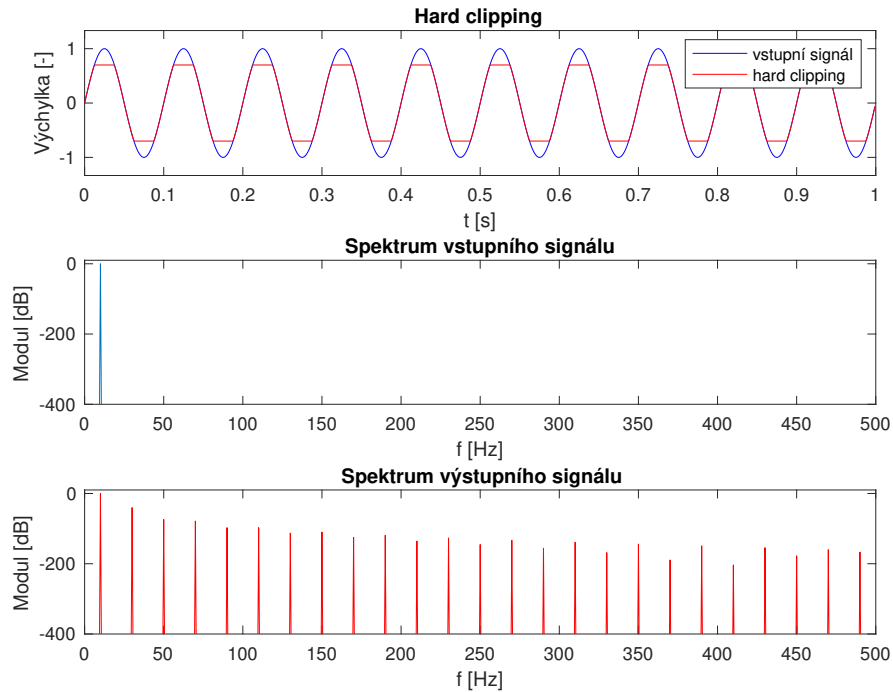
## 1.1 Hard clipping

Hard clipping je popsán následující funkcí, jejíž průběh je zachycen na obrázku 1.1:

$$x_c[n] = \begin{cases} x[n] & \text{pro } |x[n]| < \tau, \\ \tau \cdot \text{sgn}(x[n]) & \text{pro } |x[n]| \geq \tau, \end{cases} \quad (1.1)$$

kde  $x[n]$  je nepoškozený signál,  $x_c[n]$  je poškozený signál a  $\tau$  je práh (*threshold*), tedy absolutní hodnota maximální možné úrovně vzorku, nad níž budou vzorky poškozeny. V praxi může být v některých případech hodnota horní hranice  $\tau^+$  jiná, než dolní  $\tau^-$ . V této práci však budu vždy uvažovat převodní funkci symetrickou podle počátku souřadnicového systému.

Hlavním důsledkem clippingu je vznik vyšších harmonických složek ve spektru signálu. Vlastnosti těchto složek budou ovlivněny tvarem přenosové funkce signálu.



Obr. 1.2: Hard clipping – průběh signálu a porovnání spekter.

Jak můžeme pozorovat na obrázku 1.2, na kterém je původní nezkreslený signál o kmitočtu  $f = 10$  Hz, signál s ořezanými špičkami podle rovnice (1.1) pro  $\tau = 0,7$  a spektra obou signálů, při tvrdém oříznutí špiček signálu vstupního sinusového signálu se bude výstupní signál mírně podobat signálu obdelníkovému. Díky tomu vzniknou ve spektru výstupního signálu liché harmonické složky, jejichž amplitudy klesají jen velmi pozvolna. Dojde tedy k výraznému harmonickému zkreslení.

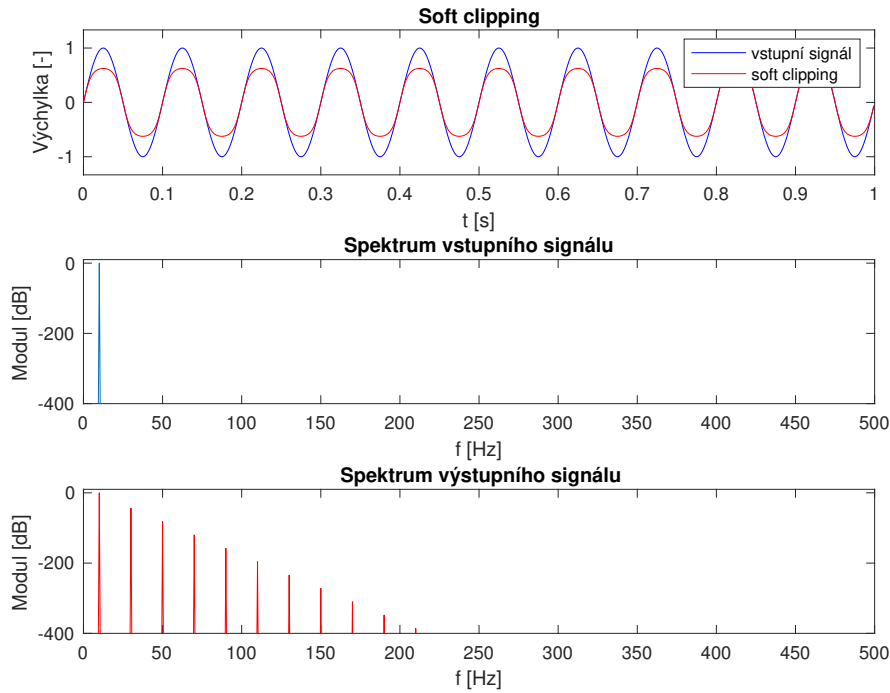
## 1.2 Soft clipping

Převodní funkce pro soft clipping, narozdíl od převodní funkce hard clippingu, není jasně dána a závisí na dané aplikaci nebo na konkrétním výrobcí. Pro převodní charakteristiky

na obrázku 1.1 byla použita následující funkce:

$$x_c[n] = \begin{cases} x[n] & \text{pro } |x[n]| \leq r, \\ \text{sgn}(x[n]) \left[ (\tau - r) \tanh\left(\frac{|x[n]| - r}{\tau - r}\right) + r \right] & \text{pro } |x[n]| > r. \end{cases} \quad (1.2)$$

Parametr  $r \in \langle 0; \tau \rangle$  určuje „tvrdost“ převodní charakteristiky. Čím vyšší hodnoty bude parametr nabývat, tím více se bude převodní funkce soft clippingu blížit funkci hard clippingu. V případě  $r = \tau$  dojde k hard clippingu, naopak v případě  $r = 0$  bude křivku převodní charakteristiky tvořit pouze funkce hyperbolický tangens. Clipping bude v tomto případě „nejměkčí“ a nejméně patrný při poslechu.



Obr. 1.3: Soft clipping – průběh signálu a porovnání spekter.

Na obrázku 1.3 můžeme pozorovat opět původní nezkreslený signál se stejnými parametry jako na obrázku 1.2, který je ořezaný podle funkce (1.2) s  $\tau = 0,7$  a s parametrem  $r = 0$ . Ve spektru ořezaného signálu můžeme pozorovat poměrně strmý pokles úrovně vyšších harmonických složek v porovnání se spektrem signálu při hard clippingu na obrázku 1.2. Pokud bychom parametr  $r$  zvolili větší než 0, bude pokles úrovně vyšších harmonických složek mnohem pozvolnější.

## 1.3 Důsledky clippingu

### 1.3.1 Harmonické zkreslení

Jak bylo zmíněno výše, hlavním důsledkem clippingu je vznik velkého množství vyšších harmonických složek ve spektru signálu.

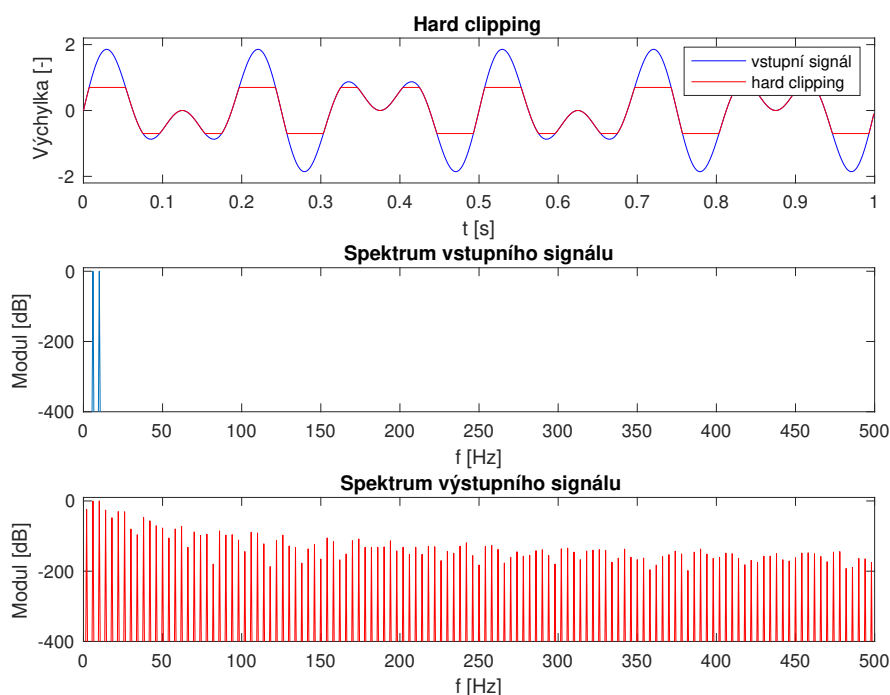


### 1.3.2 Aliasing

Jak již bylo řečeno, clipping produkuje velké množství harmonického zkreslení. Především při hard clippingu dochází ke vzniku teoreticky nekonečně mnoha vyšších harmonických složek. To znamená, že se harmonické složky nad hranici vzorkovacího kmitočtu mohou zrcadlit ve slyšitelném spektru signálu. Podle Shannon-Nyquistova teorému, musí být vzorkovací frekvence minimálně dvakrát vyšší než nejvyšší frekvence harmonických složek [4].

### 1.3.3 Intermodulační zkreslení

Intermodulační zkreslení vzniká, pokud se vstupní signál, který je následně ořezán, skládá z více harmonických složek. Kromě vzniku vyšších harmonických vznikají i složky intermodulační, které se nacházejí na součtových a rozdílových kmitočtech násobků původních harmonických složek. Na obrázku 1.4 můžeme pozorovat toto zkreslení pro signál složený ze dvou harmonických složek stejné velikosti o kmitočtu 10 Hz a 13 Hz [5].



Obr. 1.4: Intermodulační zkreslení – průběh signálu a porovnání spekter.

## 1.4 Příklady clippingu v praxi

Typicky se s clippingem setkáváme, pokud se zesilovač nachází ve stavu tzv. saturace, kde je nucen dodávat větší zesílení, než jaké mu umožňuje jeho napájecí zdroj. V takovém případě dojde ke snížení jakékoliv vyšší úrovně než je maximální možné výstupní napětí právě na tuto úroveň. Právě v zesilovačích se často setkáváme se soft clippingem, který

není při poslechu tak rušivý jako hard clipping. Speciálním případem jsou pak zesilovače pro elektrofonické kytary, u nichž se clippingu využívá k dosažení typického zkresleného zvuku.

Dalším častým příkladem clippingu je překročení dynamického rozsahu A/D převodníku během nahrávání audio signálu. Například pro rozlišení 16 bitů je nejvyšší možná kladná hodnota 32767. Pokud dojde k jejímu překročení, bude danému vzorku během procesu kvantizace přiřazena právě tato hodnota. Dojde tedy ke ztrátě dat a k oříznutí části signálu [2].

## 2 Declipping

V této kapitole se budu zabývat declippingem, tedy restaurací signálu, který byl poškozen clippingem. Popíši zde některé existující algoritmy.

### 2.1 Přehled existujících algoritmů

#### 2.1.1 Algoritmy využívající statistický model

Článek *Statistical Model-Based Approaches to Audio Restoration and Analysis* [6] z roku 2001 se zabývá nejen declippingem, ale i restaurací signálu, které jsou poničeny jinými druhy nelineárního zkreslení nebo šumem. Práce využívá statický model pro analýzu a následný odhad chybějících nebo poničených vzorků.

Na základě statistického modelu jsou odhadovány poškozené vzorky i v algoritmu, který je prezentován v článku *Monte Carlo Smoothing for Non-Linearly Distorted Signals* [7]. Oba algoritmy pochází od stejných autorů a dosahují při declippingu podobných výsledků (zlepšení poměru SNR o 8 dB při saturaci 20 % vzorků).

#### 2.1.2 Algoritmy využívající interpolaci

Prvním článkem, který se zabývá declippingem za pomoci interpolace je *Restoring a Clipped Signal* [8] z roku 1991. V tomto článku zcela chybí jakékoliv hodnocení výsledků algoritmu.

Dalším článkem je *A Simple Algorithm for the Restoration of Clipped Speech Signal* [9] z roku 2008. Ten se zabývá declippingem řečových signálů, které jsou podle autorů článku snadno předvídatelné a tudíž mohou být obnoveny s velkou přesností. Algoritmus využívá metodu lineární predikce a následné interpolace. V článku opět chybí objektivní hodnocení rekonstruovaného signálu, je však uvedeno alespoň subjektivní hodnocení pomocí poslechových testů. Posluchači hodnotili rekonstruované nahrávky jako výrazně srozumitelnější než původní satureované.

#### 2.1.3 Algoritmy využívající řídkou reprezentaci signálu

První článek, který využívá řídkou reprezentaci signálu k řešení úlohy zvané declipping je *A Constrained Matching Pursuit to Audio Declipping* [10] z roku 2011. Výstupem je algoritmus Constrained OMP (*Orthogonal Matching Pursuit*), který se často objevuje i v ostatních článcích během srovnání výsledků různých metod.

Dalším článkem je *Recovering a Clipped Signal in Sparseland* [11], který představuje dvě metody. První metoda TPCC (*Trivial Pursuit With Clipping Constraints*) využívá poznatku, že frekvence harmonických složek původního signálu odpovídají kmitočtům harmonických složek ve spektru satureovaného signálu s největší amplitudou. Původní amplituda  $k$  nejvyšších složek je odhadována pomocí metody nejmenších čtverců.

Metoda  $\text{RL}_1\text{CC}$  (*Reweighted  $\ell_1$  minimization with clipping constraints*) je iterativní metoda, která získává odhad signálu pomocí  $\ell_1$  minimalizace.

Algoritmy představeny v [11] nejsou bohužel v článku testovány na reálných signálech, ale pouze na umělých signálech složených s funkcí sinus.

Autoři článku *Consistent Iterative Hard Thresholding for Signal Declipping* [12] použili algoritmus IHT a rozšířili ho pro účely declippingu o omezující podmínky. Vzniklý algoritmus nazvaly IHT-DC (*Iterative Hard Thresholding – Declipping Constraints*). Algoritmus byl porovnán s OMP, metodou založenou na AR modelu a s komerčním software Adobe DeClipper. IHT-DC podával nejlepší výsledky, ale výsledky algoritmů byly porovnány na základě pouze dvou nahrávek.

Článek [13] z roku 2013 představuje metodu PCSL1, která využívá  $\ell_1$  minimalizaci a maskovací prahy psychoakustického modelu MPEG-1 Layer 1, kterými váhuje rekonstruovaný signál. Výsledky algoritmu jsou objektivně hodnoceny nejen pomocí poměru SNR, ale i pomocí PEAQ ODG, což je objektivní ukazatel kvality signálu podle subjektivního lidského vnímání.

Článek *Audio Declipping with Social Sparsity* [14] využívá tzv. „sociální řídkost“ pro řešení problému declippingu. Výsledný algoritmus je nejefektivnější při použití operátoru *Persistent Empirical Wiener* (PEW).

Článek *Audio Declipping by Cosparsely Hard Thresholding* [15] vychází z dříve představeného algoritmu Consistent IHT. Byl však použit analyzující model namísto syntezujícího, což vedlo k složité projekci. K řešení problému byl použit framework ADMM (*Alternating Direction Method of Multipliers*). Výsledný algoritmus CoDec-HT (*Cosparsely Declipping by Hard Thresholding*) podává, podle srovnání uvedeného v článku, lepší výsledky než původní C-IHT a srovnatelné výsledky s algoritmem Social Sparsity.

V článku *Sparsity and Cosparsity for Audio Declipping: A Flexible Non-convex Approach* [16] z roku 2015 byl představen algoritmus SPADE (*SParse Audio DEclipper*), který může být využit jak na analyzujícím, tak na syntezujícím modelu signálu (v tom případě rozlišujeme A-SPADE a S-SPADE).

Tento článek se zabývá i výpočetní náročností obou algoritmů, která závisí na volbě správného slovníku. Obecně však platí, že S-SPADE podává mírně lepší výsledky za cenu mnohem větší výpočetní náročnosti (zpracování v řádu hodin, zatímco A-SPADE zpracovává signál v řádu minut).

Článek *A Proper Version of Synthesis-based Sparse Audio Declipper* [17] z roku 2018 se zabývá úpravou algoritmu S-SPADE, přičemž verze představená v tomto článku překonává verzi z článku [16] co se kvality rekonstruovaného týče a je srovnatelná s algoritmem A-SPADE co se rychlosti a kvality rekonstrukce týče.

#### 2.1.4 Algoritmy využívající jiné metody

Mezi jiné metody můžeme zařadit například využití autoregresivního modelu, které se objevuje v článku [18] z roku 1986.

Článek [19] z roku 2014 přichází s algoritmem cBAR (*Constrained Blind Amplitude Reconstruction*), který spočívá v  $\ell_1$  minimalizaci druhé derivace vstupního signálu, což zaručuje hladkost rekonstruovaného signálu. Zajímavostí tohoto článku je způsob

hodnocení kvality rekonstrukce. Objektivním měřítkem je WER (*Word Error Rate*), což je poměr nerozpoznaných slov během automatického rozpoznání řeči.

Na tento článek navazuje článek [4], ve kterém je algoritmus upraven a tím se podstatně snižuje výpočetní náročnost. Namísto tvrdé omezující podmínky byl zaveden regularizační člen. Upravený algoritmus nese název RBAR (*Regularized Blind Amplitude Reconstruction*). Metoda RBAR zpracovává pouze znělé hlásky, které detekuje pomocí kepstrální analýzy, protože má tendenci vyhlazovat třené samohlásky.

Metoda RBAR sice nepodává tak dobré výsledky v porovnání s cBAR, ale jejím hlavním přínosem je mnohem menší výpočetní náročnost. (cBAR trvá až 400-násobek délky signálu, RBAR je průměrně  $5\times$  rychlejší než doba trvání signálu).

Článek [20] vychází z metody FSE (*Frequency Selective Extrapolation*), která se běžně používá pro restauraci nekompletních obrazových dat. V tomto článku je metoda upravena pro rekonstrukci saturovaných audio signálů. FSE iterativně generuje model signálu jako superpozici Fourierových báзовých funkcí. Saturované vzorky jsou tedy nahrazeny odhadem vytvořeným pomocí tohoto modelu. Výhodou algoritmu je především nízká výpočetní náročnost.

V roce 2015 byl publikován článek [21], který využívá faktorizaci nezápornými maticemi (NMF) pro odhad parametrů modelu a následně Wienerovu filtraci pro odhad rekonstruovaného signálu. Nejlepší výsledky, které jsou srovnatelné s výsledky algoritmu Social Sparsity, podává verze algoritmu využívající kovariační projekci.

### 3 Řídká reprezentace signálu

Tato kapitola slouží jako úvod do problematiky řídké reprezentace signálů. Nejprve se v této kapitole budu věnovat značení, poté popíši teorii bází a framů. Poslední tři podkapitoly budou věnovány řídké reprezentaci audio signálů a hledání řídkého řešení. V podkapitolách 3.1 a 3.2 jsem čerpal především z literatury [22].

#### 3.1 Základní pojmy

Skalární veličiny budou v práci značeny kurzívou (např.  $m$ ,  $N$ ), vektory budu značit malými tučnými písmenky ( $\mathbf{x}$ ,  $\mathbf{y}$ ) a matice velkými tučnými písmenky ( $\mathbf{A}$ ,  $\mathbf{B}$ ). Prvky matice budeme značit malými písmenky (např.  $a_{i,j}$ ).

Počet prvků množiny (tzv. kardinalitu) budeme značit jako absolutní hodnotu. Nosičem množiny máme na mysli množinu indexů, pro které má daný vektor nenulové hodnoty. Tuto množinu budeme značit  $\text{supp}(\mathbf{x})$ .

Pro řídký signál  $\mathbf{x} = [0, 0, 0, 1, 0, 4, 8, 4, 0, 0, 1, 0]^\top$  tedy platí následující:  $\text{supp}(\mathbf{x}) = \{4, 6, 7, 8, 11\}$  a  $|\text{supp}(\mathbf{x})| = 5$ .

Norma vektoru je míra, která jedním číslem vyjadřuje jeho „velikost“. Nejpoužívanější normou je norma euklidovská  $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$ . Pro práci s řídkou reprezentací signálu je však nutné definovat  $\ell_p$ -normu vektoru<sup>1</sup>

$$\begin{aligned} \|\mathbf{x}\|_p &:= \left( \sum_{i=1}^n |x_i|^p \right)^{1/p} && \text{pro } 1 \leq p < \infty, \\ \|\mathbf{x}\|_p &:= \sum_{i=1}^n |x_i|^p && \text{pro } 0 < p < 1, \\ \|\mathbf{x}\|_\infty &:= \max_i |x_i|, \\ \|\mathbf{x}\|_0 &:= |\text{supp}(\mathbf{x})|. \end{aligned} \tag{3.1}$$

$\ell_1$  norma (nebo také Manhattan norma) představuje součet všech absolutních hodnot vektoru, zatímco  $\ell_0$  norma počet nenulových složek vektoru. Vektor je  $k$ -řídký, pokud platí:

$$\|\mathbf{x}\|_0 \leq k. \tag{3.2}$$

Tedy aby byl vektor  $k$ -řídký, musí mít nanejvýše  $k$  nenulových členů. V praxi nemusí být vektor reprezentující reálný signál řídký ve striktním smyslu, ale místo nulových hodnot obsahuje malé nenulové hodnoty, které můžeme zanedbat. Pojmem relativní řídkost vektoru délky  $N$  myslíme poměr  $\frac{k}{N}$ .

<sup>1</sup>striktně vzato se o normu jedná jen v případě  $1 \leq p \leq \infty$ . V této práci však budu jako normu označovat i „preudonormu“  $\ell_0$ .

## 3.2 Báze a framy

### 3.2.1 Vektorový prostor

Vektorový prostor (VP) je algebraická struktura, která splňuje známé axiomy. Základními prvky vektorového prostoru jsou vektory. V této práci budu uvažovat dimenzi vektorového prostoru  $n$  jako konečné číslo, to znamená, že budu pracovat pouze s konečnou množinou těchto vektorů, kterou budeme značit jako matici, která vznikne seřazením vektorů do sloupců.

Systém generátorů vektorového prostoru  $\mathbb{V}$  je množina vektorů  $\mathbf{E} \in \mathbb{V}$ . Každý vektor  $\mathbf{x} \in \mathbb{V}$  je lineární kombinací těchto generátorů. Pokud je generátorů VP více, než je jeho dimenze  $n$ , je systém nedourčený a vektor  $\mathbf{x} \in \mathbb{V}$  lze zapsat jako různé lineární kombinace generátorů. Vektor  $\mathbf{x} \in \mathbb{V}$  lze zapsat jako:

$$\mathbf{x} = c_1 \mathbf{e}_1 + c_2 \mathbf{e}_2 + \dots + c_m \mathbf{e}_m = \mathbf{E} \mathbf{c}, \quad (3.3)$$

kde vektor  $\mathbf{c}$  je vektorem souřadnic vektoru  $\mathbf{x}$  v  $\mathbf{E}$ .

### 3.2.2 Báze vektorového prostoru

Bází vektorového prostoru je množina maximálního počtu lineárně nezávislých vektorů daného prostoru. Lineární kombinací těchto vektorů lze vytvořit libovolný vektor v daném VP. To znamená že pomocí báze je každý vektor v prostoru dán jednoznačným vektorem souřadnic  $\mathbf{c}$ .

#### Ortogonální a ortonormální báze

V praxi se díky snadnějšímu použití velmi často pracuje s bázemi ortogonálními a ortonormálními (například kvůli numerické stabilitě). Pro jakékoliv dva vektory ortogonální báze  $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$  platí:

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0, \quad \langle \mathbf{b}_i, \mathbf{b}_i \rangle \neq 0. \quad (3.4)$$

To znamená, že jsou na sebe každé dva vektory báze kolmé. Pro všechny vektory ortonormální báze kromě vzorce (3.4) platí i  $\|\mathbf{b}_i\| = 1$ .

### 3.2.3 Framy

Jak vyplývá z předchozích odstavců, bázi konečněrozměrného vektorového prostoru  $\mathbb{V}$  tvoří lineárně nezávislé vektory a jejich počet je roven dimenzi prostoru  $n$ . Pokud je generátorů více, než jaký je rozměr dimenze prostoru (nutně jsou tedy lineárně závislé), tvoří tzv. *frame* ve  $\mathbb{V}$ .

Tato nadbytečnost může být pro (obzvláště řídkou) reprezentaci signálu výhodou. Nevýhodou je riziko numerické nestability.

Aby nadbytečná množina generátorů  $\{\phi_k\}_{k \in \mathbb{I}}$  tvořila frame, musí existovat konstanty  $0 < A \leq B < \infty$ , pro které platí:

$$A\|\mathbf{x}\|^2 \leq \sum_{k \in \mathbb{I}} |\langle \mathbf{x}, \phi_k \rangle|^2 \leq B\|\mathbf{x}\|^2, \quad \forall \mathbf{x} \in \mathbb{V}. \quad (3.5)$$

Prvky framu  $\phi_k$  nazýváme *atomy*, konstanty  $A, B$  nazýváme *mezemi framu*. Pokud platí  $A = B$ , nazýváme frame *těsným*.

### 3.3 Řídká reprezentace audio signálu

Audio signály, jako je řeč nebo hudba, jsou řídké v časové nebo frekvenční oblasti. Zatímco řídkost v časové oblasti se pojí s přechodovými jevy signálu, ve frekvenční oblasti můžeme pozorovat řídkou reprezentaci v podobě harmonických složek signálu.

Audio signál lze například pomocí Fourierovy transformace vyjádřit jako součin transformační matice, kterou tvoří jednotlivé harmonické složky, a vektoru koeficientů. Rozlišujeme buď analyzující nebo syntezující model transformace.

Běžnější je použití syntezujícího modelu, při kterém pro libovolný signál  $\mathbf{x}$  platí:  $\mathbf{x} = \mathbf{D}\mathbf{z}$ , kde  $\mathbf{D}$  je slovník (transformační matice) a  $\mathbf{z}$  je řídký vektor koeficientů transformace. Například při použití Fourierovy transformace vzniká výsledný signál syntézou jednotlivých harmonických složek obsažených ve slovníku  $\mathbf{D}$  [23].

Při použití analyzujícího modelu vznikne aplikací analyzujícího operátoru  $\mathbf{A}$  řídký vektor koeficientů  $\mathbf{z}$  ( $\mathbf{A}\mathbf{x} = \mathbf{z}$ ).

Oba modely jsou shodné, pokud jsou jejich transformační matice  $\mathbf{A}$  a  $\mathbf{D}$  ortonormální báze ( $\mathbf{D} = \mathbf{A}^{-1}$ ).

### 3.4 Strukturovaná řídkost

Jak je uvedeno v [22], v praktických aplikacích při hledání řídkého řešení se často koeficienty vyskytují ve skupinách. Pokud například bude nějaký koeficient nenulový, je velmi pravděpodobné, že jiný konkrétní koeficient bude také nenulový. Při hledání řídkého řešení nás tedy nezajímají jednotlivé koeficienty, ale celé skupiny koeficientů, které mezi sebou mají určité vazby.

Signálem, který je skupinově řídký, je myšlen signál, který je možno reprezentovat pomocí malého množství skupin koeficientů.

Podle vzájemných vztahů mezi skupinami a koeficienty můžeme strukturovanou řídkost dále dělit na:

- *Simultánní řídkost* - zvláštní případ skupinové řídkosti, kdy dopředu víme, že několik pozorovaných signálů má stejnou strukturu řídkosti.
- *Elitářskou řídkost* - stav, kdy jsou všechny skupiny významné, ale v rámci skupiny je vyžadován pouze malý počet významných koeficientů
- *Sociální řídkost* - nastává, pokud jeden koeficient spadá do více skupin koeficientů

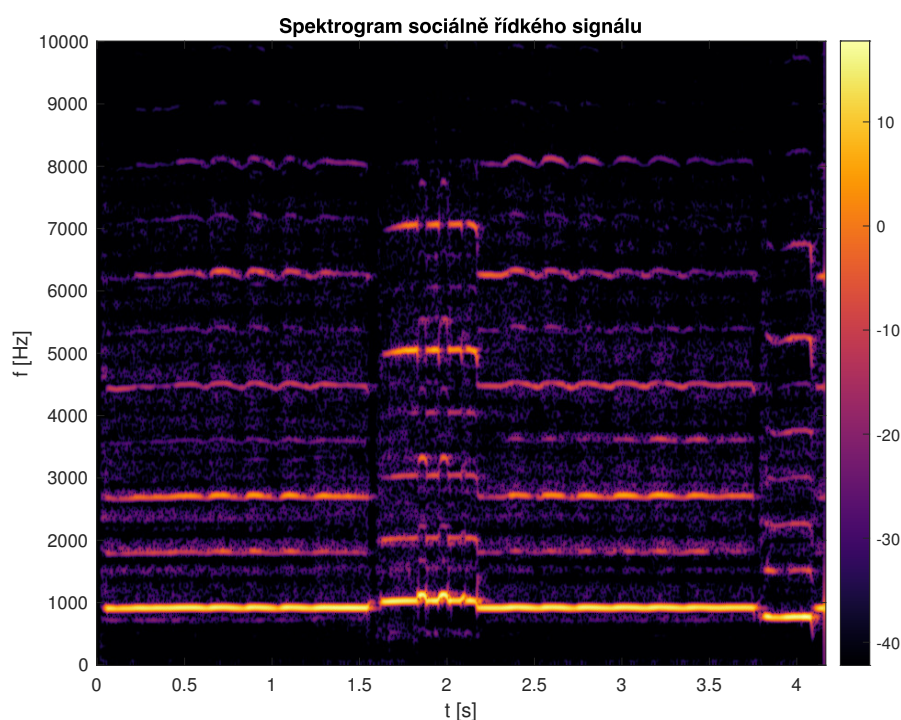


- *Hierarchickou řídkost* - situace, kdy existují vztahy mezi jednotlivými skupinami koeficientů

### 3.4.1 Strukturovaná řídkost v audio signálech

Příklad skupinově řídkého audio signálu můžeme pozorovat na obrázku 3.1, kdy je spektrum signálu tvořeno fundamentem a od něj odvozenými vyššími harmonickými složkami, jejichž kmitočet je celočíselným násobkem kmitočtu fundamentu.

Vzhledem k faktu, že většinu audio signálů lze považovat za stacionární jen ve velmi krátkém časovém úseku (asi 30 ms), má smysl tonální skupiny koeficientů vytvářet maximálně takto dlouhé.



Obr. 3.1: Spektrogram sociálně řídkého signálu.

## 3.5 Hledání řídkého řešení

Nalezt přesné řídké řešení optimalizační úlohy není vždy možné, proto se v praxi k jeho hledání používají aproximativní metody, které můžeme zjednodušeně rozdělit do tří kategorií.

### 3.5.1 Hladové algoritmy

Při použití hladového algoritmu je v každé iteraci nalezen nejvýznamnější atom, který už nemůže být zbaven na konečném výsledku. Nevýhodou těchto algoritmu je možnost

nenalezení globálního optima. Výhodou nízká výpočetní náročnost. Do této kategorie spadají například algoritmy MP (*Matching Pursuit*) nebo OMP (*Orthogonal Matching Pursuit*).

### 3.5.2 Relaxační algoritmy

Tyto algoritmy jsou založeny na  $\ell_1$ -relaxaci. Mezi tyto algoritmy patří například  $\text{R}\ell_1\text{CC}$  nebo PCSL1 z kapitoly 2.1.3. Na řešení těchto úloh se využívají proximální algoritmy. Optimalizace pomocí těchto algoritmů je založena na iterativní minimalizaci, přičemž jsou známy podmínky, za kterých algoritmus konverguje k řešení úlohy. Výhodou těchto algoritmů je jejich flexibilita, která je však vykoupena nízkou rychlostí výpočtu v porovnání s některými specializovanými algoritmy.

### 3.5.3 Hybridní algoritmy

Mimo tyto dvě kategorie spadají tzv. hybridní algoritmy, které využívají přednosti obou skupin algoritmů. Mezi tyto metody spadají algoritmy založené na prahování.

## ADMM

V této části čerpám především z literatury [17]. Nalezení řídkého řešení, které by odpovídalo všem podmínkám, může být velmi obtížné (*NP-Hard*). K jeho nalezení se často využívá metoda ADMM (*Alternating Direction Method of Multipliers*), která se používá k řešení minimalizačních úloh duálních proměnných. Typická rovnice řešená pomocí ADMM lze zapsat jako:

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{Ax}) \quad (3.6)$$

kde  $f$  a  $g$  jsou funkce reálných (případně komplexních) proměnných. Rovnice (3.6) může být také formulována jako:

$$\min_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \quad \text{s.t.} \quad \mathbf{Ax} - \mathbf{z} = 0. \quad (3.7)$$

K řešení rovnice (3.7) je sestaven tzv. *Augmented Lagrangian*:

$$L_\rho(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^\top (\mathbf{Ax} - \mathbf{z}) + \frac{\rho}{2} \|\mathbf{Ax} - \mathbf{z}\|_2^2, \quad (3.8)$$

kde  $\rho$  je tzv. *penalizační parametr*. Po převedení rovnice (3.8) do tzv. *scaled form*, tedy nahrazení výrazu  $\mathbf{Ax} - \mathbf{z}$  reziduem  $\mathbf{r}$  a po zavedení duální proměnné  $\mathbf{u} = \frac{1}{\rho} \mathbf{y}$  lze tuto rovnici převést na tvar:

$$L_\rho(\mathbf{x}, \mathbf{u}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{r} + \mathbf{u}\|_2^2 - \frac{\rho}{2} \|\mathbf{u}\|_2^2 \quad (3.9)$$

Následně můžeme iterativně postupnou aktualizací proměnných  $\mathbf{x}$ ,  $\mathbf{z}$  a duální proměnné  $\mathbf{u}$  najít řídké řešení:

$$\mathbf{x}^{i+1} = \arg \min_x \left( f(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{Ax} - \mathbf{z}^i + \mathbf{u}^i\|_2^2 \right) \quad (3.10a)$$

$$\mathbf{z}^{i+1} = \arg \min_x \left( g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{Ax}^{i+1} - \mathbf{z} + \mathbf{u}^i\|_2^2 \right) \quad (3.10b)$$

$$\mathbf{u}^{i+1} = \mathbf{u}^i + \mathbf{Ax}^{i+1} - \mathbf{z}^{i+1}. \quad (3.10c)$$

Při minimalizaci rovnice (3.9) podle  $\mathbf{x}$  (rovnice (3.10a)) můžeme vynechat člen  $g(\mathbf{z})$ , protože se nijak nepodílí na nalezení argumentu minima. Totéž platí o členu  $f(\mathbf{x})$  v rovnici (3.10b) a členu  $\frac{\rho}{2} \|\mathbf{u}\|_2^2$  v obou rovnicích.

## 4 Hledání řídkého řešení v praxi

V této kapitole se budu věnovat dvěma algoritmům, které řeší úlohu declippingu. Oba tyto algoritmy jsou zaměřeny na tvrdě ořezané signály podle vzorce (1.1). Prvním z nich je algoritmus SPADE, představený v článku [16], který se věnuje rekonstrukci monofonních signálů. Algoritmus CASCADE, prezentovaný v článku [29], je v podstatě rozšířením původního algoritmu SPADE, který je upraven pro rekonstrukci vícekanálových signálů. Nejdříve však v této kapitole popíši segmentaci signálu, pomocí které je třeba signál rozdělit na bloky ještě před rekonstrukcí a rozdělení vzorků signálu na spolehlivé a saturované.

### 4.1 Segmentace signálu

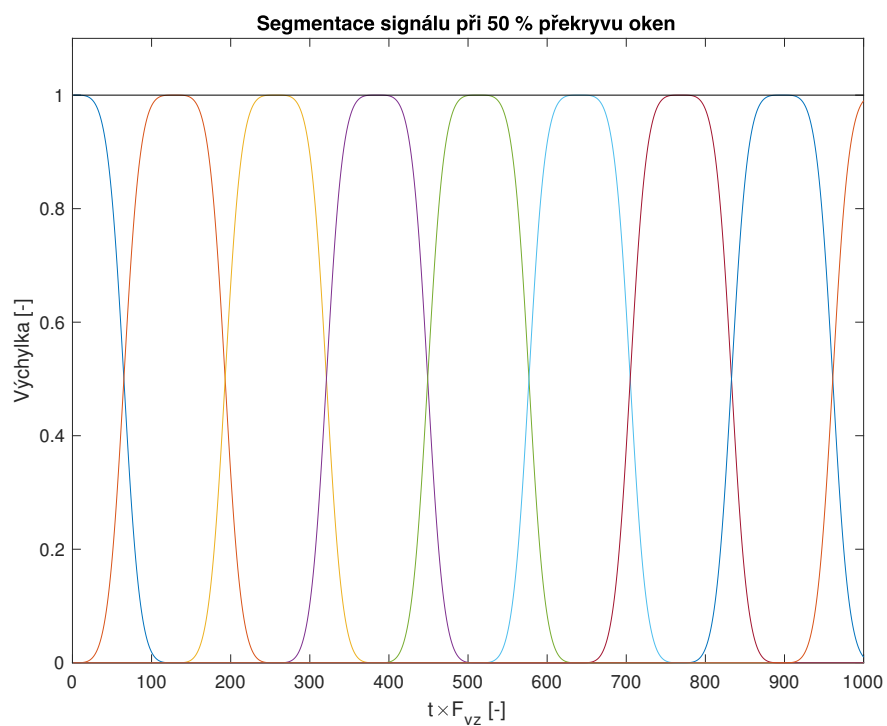
Jak již bylo řečeno, oba rekonstrukční algoritmy zpracovávají signál po částech, přičemž pro segmentaci používají metodu OLA (*Overlap-add*). Signál je během segmentace rozdělen do překrývajících se bloků. Tyto vzorky jsou následně váhovány, nejčastěji Hannovým oknem. Po rekonstrukci jsou jednotlivé bloky opět sečteny do výsledného signálu. V této části jsem čerpal z literatury [25], [26], [27] a [28].

Na obrázcích 4.1 a 4.2 jsou zobrazena Hannova okna, pomocí kterých je signál váhován. Délka okna je 256 vzorků a překryv oken je 50 %, respektive 75 % (tedy posun mezi sousedními okny je 128 a 64 vzorků). Okna jsou normalizována tak, že po sečtení jednotlivých segmentů má signál stejnou amplitudu jako před segmentací. Z grafů můžeme pozorovat, že na rekonstrukci jednoho vzorku saturovaného signálu se podílí součet rekonstruovaných signálů ve dvou oknech v případě 50% překryvu a ve čtyřech oknech v případě 75% překryvu. To znamená, že pokud v jednom bloku dojde k chybě při rekonstrukci konkrétního vzorku, v dalším okně mohou být následky této chyby zmírněny.

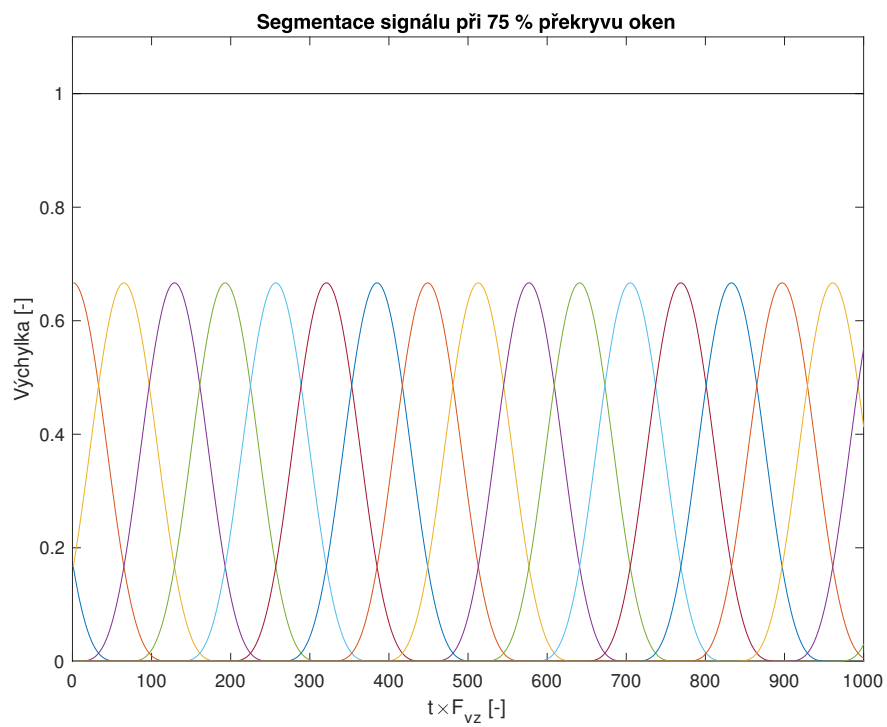
### 4.2 Rozdělení vzorků saturovaného signálu

Díky tvrdému ořezání můžeme vzorky signálu snadno rozdělit na množinu spolehlivých (*reliable*) vzorků, které nebyly poškozeny clippingem a na dvě množiny saturovaných vzorků (*clipped*) s kladnou a se zápornou výchylnou pomocí masek  $\mathbf{M}_r$ ,  $\mathbf{M}_c^+$  a  $\mathbf{M}_c^-$ .

Každá z masek má stejnou délku jako má signál. Maska má na každé pozici buď hodnotu 1 nebo 0 podle toho, zda vzorek spadá do dané množiny. Například maska  $\mathbf{M}_r$  má na všech pozicích spolehlivých vzorků hodnotu 1, na pozicích saturovaných vzorků má hodnotu 0.



Obr. 4.1: Hannova okna s překryvem 50 %.



Obr. 4.2: Hannova okna s překryvem 75 %.

Pro odhad  $\hat{\mathbf{x}}$  původního nesaturovaného signálu  $\mathbf{x}$  musí platit následující podmínka:

$$\mathbf{M}_r \hat{\mathbf{x}} = \mathbf{M}_r \mathbf{y} \quad \mathbf{M}_c^+ \hat{\mathbf{x}} \geq \mathbf{M}_c^+ \mathbf{y} \quad \mathbf{M}_c^- \hat{\mathbf{x}} \leq \mathbf{M}_c^- \mathbf{y} \quad (4.1)$$

Kterou zjednodušeně zapisujeme jako  $\hat{\mathbf{x}} \in \Gamma(\mathbf{y})$ , kde množinu  $\Gamma$  zapisujeme jako:

$$\Gamma(\mathbf{y}) = \{\hat{\mathbf{x}} \mid \mathbf{M}_r \hat{\mathbf{x}} = \mathbf{M}_r \mathbf{y}, \mathbf{M}_{c+} \hat{\mathbf{x}} \geq \tau, \mathbf{M}_{c-} \hat{\mathbf{x}} \leq -\tau\}. \quad (4.2)$$

### 4.3 Algoritmus SPADE

SPADE je heuristický algoritmus, který může být použit pro rekonstrukci saturovaného signálu jak pomocí analyzujícího, tak syntezujícího modelu signálu. Odhad saturovaného signálu je v případě analyzujícího respektive syntezujícího modelu získán vyřešením optimalizačního problému:

$$\min_{\mathbf{x}, \mathbf{z}} \|\mathbf{z}\|_0 + \mathbf{1}_{\Gamma(\mathbf{y})}(\mathbf{x}) + \mathbf{1}_{\ell_2 \leq \varepsilon}(\mathbf{x} - \mathbf{D}\mathbf{z}) \quad (4.3)$$

$$\min_{\mathbf{x}, \mathbf{z}} \|\mathbf{z}\|_0 + \mathbf{1}_{\Gamma(\mathbf{y})}(\mathbf{x}) + \mathbf{1}_{\ell_2 \leq \varepsilon}(\mathbf{A}\mathbf{x} - \mathbf{z}) \quad (4.4)$$

Kde  $\mathbf{1}_{\Gamma(\mathbf{y})}(\mathbf{x})$  je indikátorová funkce, vynucující podmínku (4.1) a  $\mathbf{1}_{\ell_2 \leq \varepsilon}$  je indikátorová funkce, která minimalizuje  $\ell_2$  normu argumentu funkce, což je rozdíl mezi minimalizovanými proměnnými  $\mathbf{x}$  a  $\mathbf{z}$ , který v případě syntezujícího modelu probíhá v časové oblasti pomocí operátoru  $\mathbf{D}$  a v případě analyzujícího modelu v oblasti koeficientů pomocí operátoru  $\mathbf{A}$ .

#### ADMM

Řešení tohoto problému je však velmi obtížné (*NP-Hard*). K nalezení přibližného řešení je možné použít metodu ADMM představenou v části 3.5.3. Optimalizační problémy (4.3) a (4.4) je tedy možné vyřešit iterativně pomocí tří následujících kroků:

<b>A-SPADE</b>	<b>S-SPADE</b>
$\mathbf{z}^{(i+1)} = \arg \min_{\mathbf{z}} \ \mathbf{A}\mathbf{x}^{(i)} - \mathbf{z} + \mathbf{u}^{(i)}\ _2^2$ $\text{s.t. } \ \mathbf{z}\ _0 \leq k$ $\mathbf{x}^{(i+1)} = \arg \min_{\mathbf{x}} \ \mathbf{A}\mathbf{x} - \mathbf{z}^{(i+1)} + \mathbf{u}^{(i)}\ _2^2$ $\text{s.t. } \mathbf{x} \in \Gamma$ $\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \mathbf{A}\mathbf{x}^{(i+1)} - \mathbf{z}^{(i+1)}. \quad (4.5c)$	$\mathbf{z}^{(i+1)} = \arg \min_{\mathbf{z}} \ \mathbf{D}\mathbf{z} - \mathbf{x}^{(i)} + \mathbf{u}^{(i)}\ _2^2$ $\text{s.t. } \ \mathbf{z}\ _0 \leq k$ $\mathbf{x}^{(i+1)} = \arg \min_{\mathbf{x}} \ \mathbf{D}\mathbf{z}^{(i+1)} - \mathbf{x} + \mathbf{u}^{(i)}\ _2^2$ $\text{s.t. } \mathbf{x} \in \Gamma$ $\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \mathbf{D}\mathbf{z}^{(i+1)} - \mathbf{x}^{(i+1)}. \quad (4.6c)$
(4.5a)	(4.6a)
(4.5b)	(4.6b)

První krok algoritmu A-SPADE (4.5a) lze vyřešit pomocí funkce hard thresholding, který byl pro účely declippingu poprvé použit v článku [12]. Pomocí té stejné funkce je možné aproximovat také první krok algoritmu S-SPADE (4.6a). Horní index  $(i)$  značí i-tou iteraci.

## Hard Thresholding

Funkce hard thresholding ( $\mathcal{H}_k$ ), vybere ze spektra signálu  $k$  nejvyšších koeficientů, ostatní jsou zanedbány. Problémem je odhadnout velikost  $k$  předem. Algoritmus proto pracuje iterativně, kdy je na začátku nastavena nějaká malá hodnota  $k$  a algoritmus (4.5) (respektive (4.6)) probíhá opakovaně pro rostoucí  $k$  tak dlouho, dokud není signál dostatečně zrekonstruován.

## Projekce na přípustnou množinu řešení

Projekce na přípustnou množinu řešení je druhým krokem algoritmu ((4.5b) a (4.6b)). Signál je po hard thresholdingu z frekvenční domény převeden zpět do časové a je provedena projekce na množinu  $\Gamma(\mathbf{y})$ .

V praxi to znamená, že rekonstruované vzorky na spolehlivých pozicích  $\mathbf{M}_r$  jsou nahrazeny původními vzorky saturovaného signálu. Pokud je hodnota vzorku na kladné saturované pozici  $\mathbf{M}_c^+$  menší než práh  $\tau$ , je mu přiřazena hodnota tohoto prahu. Pokud je hodnota vzorku na této pozici stejná nebo vyšší než hodnota  $\tau$ , je mu tato hodnota ponechána. Obdobně probíhá projekce samozřejmě i na záporných saturovaných pozicích  $\mathbf{M}_c^-$ .

---

### Algorithm 1 A-SPADE

---

**Require:**  $\mathbf{A}, \mathbf{y}, \mathbf{M}_r, \mathbf{M}_c^+, \mathbf{M}_c^-, s, r, \varepsilon$

```

1:  $\hat{\mathbf{x}}^{(0)} = \mathbf{y}, \mathbf{u}^{(0)} = 0, i = 1, k = s$ 
2:  $\bar{\mathbf{z}}^{(i)} = \mathcal{H}_k(\mathbf{A}\hat{\mathbf{x}}^{(i-1)} + \mathbf{u}^{(i-1)})$ 
3:  $\hat{\mathbf{x}}^{(i)} = \arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \bar{\mathbf{z}}^{(i)} + \mathbf{u}^{(i-1)}\|_2^2$ 
   s.t.  $\mathbf{x} \in \Gamma$ 
4: if  $\|\mathbf{A}\hat{\mathbf{x}}^{(i)} - \bar{\mathbf{z}}^{(i)}\|_2 \leq \varepsilon$  then
5:   terminate
6: else
7:    $\mathbf{u}^{(i)} = \mathbf{u}^{(i-1)} + \mathbf{A}\hat{\mathbf{x}}^{(i)} - \bar{\mathbf{z}}^{(i)}$ 
8:    $i \leftarrow i + 1$ 
9:   if  $i \bmod r = 0$  then
10:     $k \leftarrow k + s$ 
11:   end if
12:   go to 2
13: end if
14: return  $\hat{\mathbf{x}} = \hat{\mathbf{x}}^{(i)}$ 
```

---



---

### Algorithm 2 S-SPADE

---

**Require:**  $\mathbf{D}, \mathbf{y}, \mathbf{M}_r, \mathbf{M}_c^+, \mathbf{M}_c^-, s, r, \varepsilon$

```

1:  $\hat{\mathbf{x}}^{(0)} = \mathbf{y}, \mathbf{u}^{(0)} = 0, i = 1, k = s$ 
2:  $\bar{\mathbf{z}}^{(i)} = \mathcal{H}_k(\mathbf{D}^*(\hat{\mathbf{x}}^{(i-1)} - \mathbf{u}^{(i-1)}))$ 
3:  $\hat{\mathbf{x}}^{(i)} = \arg \min_{\mathbf{x}} \|\mathbf{D}\bar{\mathbf{z}}^{(i)} - \mathbf{x} + \mathbf{u}^{(i-1)}\|_2^2$ 
   s.t.  $\mathbf{x} \in \Gamma$ 
4: if  $\|\mathbf{D}\bar{\mathbf{z}}^{(i)} - \hat{\mathbf{x}}^{(i)}\|_2 \leq \varepsilon$  then
5:   terminate
6: else
7:    $\mathbf{u}^{(i)} = \mathbf{u}^{(i-1)} + \mathbf{D}\bar{\mathbf{z}}^{(i)} - \hat{\mathbf{x}}^{(i)}$ 
8:    $i \leftarrow i + 1$ 
9:   if  $i \bmod r = 0$  then
10:     $k \leftarrow k + s$ 
11:   end if
12:   go to 2
13: end if
14: return  $\hat{\mathbf{x}} = \hat{\mathbf{x}}^{(i)}$ 
```

---

## Implementace

Poslední třetí krok algoritmu (4.5c) (a (4.6c)) je aktualizace duální proměnné  $\mathbf{u}$ . Tyto tři kroky tvoří základ rekonstrukčního algoritmu SPADE. Finální podoba obou rekonstrukčních algoritmů je uvedena v algoritmech 1 a 2. Velmi důležitým krokem je

podmínka, která ukončí iterativní algoritmus, pokud je dosaženo dostatečně malého rozdílu mezi signálem před projekcí a po projekci.

Parametr  $r$  udává, po kolika iteracích se počet  $k$  nejvyšších koeficientů zvýší o  $s$ . Parametr  $\varepsilon$  ovlivňuje požadovanou přesnost algoritmu.

Vypočetní náročnost algoritmu SPADE závisí mimo jiné také na volbě transformační matice  $\mathbf{A}$  nebo slovníku  $\mathbf{D}$ .

## 4.4 Algoritmus CASCADE

Algoritmus CASCADE (*Channel-Aware Structured Cospase Audio Declipper*) byl navržen pro rekonstrukci saturevaných vícekanálových audio signálů. K řešení tohoto problému využívá základy algoritmu A-SPADE a strukturované řídkosti vícekanálových signálů. Konkrétně tento algoritmus využívá simultánní řídkosti všech kanálů a skupinové řídkosti napříč kanály.<sup>1</sup>

Základem algoritmu je předpoklad, že nenulové koeficienty jsou v jednotlivých kanálech signálu obsaženy rovnoměrně. Tím pádem k rekonstrukci signálu stačí najít dostatečný počet nejvýznamnějších skupin.

Signál byl u algoritmu SPADE reprezentován vektorem (ať už v časové nebo frekvenční oblasti). U algoritmu CASCADE je signál v obou doménách reprezentován maticemi, jejichž jeden rozměr odpovídá délce signálu a druhý rozměr počtu kanálů.

### Shrinkage operátor GEW

Algoritmus CASCADE využívá k nalezení těchto nejvýznamnějších skupin shrinkage operátor s názvem *Group Empirical Wiener* (GEW), který plní podobnou funkci jako hard thresholding u algoritmu SPADE.

$$\mathcal{S}_\mu(\mathbf{Z})_{pk} = \mathbf{Z}_{pk} \left( 1 - \frac{\mu^2}{\|\mathbf{z}_p\|_2^2} \right)_+ \quad (4.7)$$

$\mathbf{Z}$  je matice koeficientů, její první index  $p$  udává kmitočet koeficientu a index  $k$  kanál koeficientu. Parametr  $\mu$  ovlivňuje citlivost prahování. Znaménko  $+$  v indexu závorky značí funkci  $(a) = \max(a; 0)$ .

GEW operátor v podstatě každý koeficient vynásobí vahou, která závisí na velikosti daného koeficientu ve všech kanálech. Pokud není energie koeficientu ve všech kanálech dostatečně velká, dojde k vynásobení nulou. To, jaká hodnota energie bude považována za dostatečně velkou, ovlivňuje právě parametr  $\mu$ , který je v každé iteraci algoritmu CASCADE zmenšen.

<sup>1</sup>Autoři algoritmu představili pouze analyzující verzi algoritmu a tu nazvali CASCADE. Syntezující verze by měla být správně podle logiky věci nazývána CASSADE (*Channel-Aware Structured Sparse Audio Declipper*). Pro přehlednost ale budu v této práci rozlišovat analyzující a syntezující verzi algoritmu jako A-CASCADE a S-CASCADE, stejně jako je tomu u algoritmu SPADE.



---

**Algorithm 3** A-CASCADE

---

**Require:**  $\mathbf{A}, \mathbf{Y}, \mathbf{M}_r, \mathbf{M}_c^+, \mathbf{M}_c^-, \alpha, \beta, \mu^{(0)}$ 

```
1:  $\hat{\mathbf{X}}^{(0)} = \mathbf{Y}, \mathbf{U}^{(0)} = 0, i = 1$ 
2:  $\bar{\mathbf{Z}}^{(i)} = \mathcal{S}_{\mu^{(i-1)}}(\mathbf{A}\hat{\mathbf{X}}^{(i-1)} + \mathbf{U}^{(i-1)})$ 
3:  $\hat{\mathbf{X}}^{(i)} = \arg \min_{\mathbf{X}} \|\mathbf{A}\mathbf{X} - \bar{\mathbf{Z}}^{(i)} + \mathbf{U}^{(i-1)}\|_2^2$ 
   s.t.  $\mathbf{X} \in \Gamma$ 
4: if  $\frac{\|\mathbf{A}\hat{\mathbf{X}}^{(i)} - \bar{\mathbf{Z}}^{(i)}\|_2}{\|\mathbf{A}\hat{\mathbf{X}}^{(i)}\|_2} \leq \beta$  then
5:   terminate
6: else
7:    $\mu^{(i)} = \alpha\mu^{(i-1)}$ 
8:    $\mathbf{U}^{(i)} = \mathbf{U}^{(i-1)} + \mathbf{A}\hat{\mathbf{X}}^{(i)} - \bar{\mathbf{Z}}^{(i)}$ 
9:    $i \leftarrow i + 1$ 
10:  go to 2
11: end if
12: return  $\hat{\mathbf{X}} = \hat{\mathbf{X}}^{(i)}$ 
```

---

---

**Algorithm 4** S-CASCADE

---

**Require:**  $\mathbf{D}, \mathbf{Y}, \mathbf{M}_r, \mathbf{M}_c^+, \mathbf{M}_c^-, \alpha, \beta, \mu^{(0)}$ 

```
1:  $\hat{\mathbf{X}}^{(0)} = \mathbf{Y}, \mathbf{U}^{(0)} = 0, i = 1$ 
2:  $\bar{\mathbf{Z}}^{(i)} = \mathcal{S}_{\mu^{(i-1)}}(\mathbf{D}^*(\hat{\mathbf{X}}^{(i-1)} - \mathbf{U}^{(i-1)}))$ 
3:  $\hat{\mathbf{X}}^{(i)} = \arg \min_{\mathbf{X}} \|\mathbf{D}\bar{\mathbf{Z}}^{(i)} - \mathbf{X} + \mathbf{U}^{(i-1)}\|_2^2$ 
   s.t.  $\mathbf{X} \in \Gamma$ 
4: if  $\frac{\|\mathbf{D}\bar{\mathbf{Z}}^{(i)} - \hat{\mathbf{X}}^{(i)}\|_2}{\|\hat{\mathbf{X}}^{(i)}\|_2} \leq \beta$  then
5:   terminate
6: else
7:    $\mu^{(i)} = \alpha\mu^{(i-1)}$ 
8:    $\mathbf{U}^{(i)} = \mathbf{U}^{(i-1)} + \mathbf{D}\bar{\mathbf{Z}}^{(i)} - \hat{\mathbf{X}}^{(i)}$ 
9:    $i \leftarrow i + 1$ 
10:  go to 2
11: end if
12: return  $\hat{\mathbf{X}} = \hat{\mathbf{X}}^{(i)}$ 
```

---

Kromě parametru  $\mu$  jsou pro algoritmus CASCADE zásadní ještě parametry  $\alpha$  a  $\beta$ . Parametrem  $\alpha \in \langle 0; 1 \rangle$  je v každém kromu vynásoben parametr  $\mu$ . Ten tedy v podstatě udává, jak „agresivně“ bude algoritmus postupovat. Čím je  $\alpha$  menší, tím více harmonických složek je v každém kroku vybráno. Podle autorů funguje algoritmus nejlépe pro  $\alpha = 0,99$ .

Parametr  $\beta$  ovlivňuje požadovanou přesnost algoritmu, je to v podstatě obdoba parametru  $\varepsilon$  u algoritmu SPADE.

## 5 Algoritmus

V této kapitole se věnuji implementaci algoritmů SPADE a CASCADE, popsaným v předchozí kapitole, kterou jsem provedl v prostředí Matlab, konkrétně ve verzi 2018b. K implementaci je využit toolbox LTFAT (*The Large Time/Frequency Analysis Toolbox*) [30], který umožňuje efektivně pracovat s analýzou a syntézou v časově-frekvenční oblasti. Toolbox je zdarma ke stažení na stránkách <http://ltfat.github.io>.

Algoritmus tvoří dohromady deset m-souborů: hlavní soubor `declipping.m`, soubor `segmentation.m`, který obsahuje funkci provádějící segmentaci. Se svolením autorů jsem implementaci této funkce převzal z článku [17] a upravil ji podle svých potřeb. Další čtyři soubory tvoří jednotlivé algoritmy (`ASPADE.m`, `SSPADE.m`, `ACASCADE.m` a `SCASCADE.m`). Dále algoritmus obsahuje čtyři soubory s pomocnými funkcemi (`clipping.m`, `projection.m`, `hard_thresholding.m` a `sdr_func.m`).

### 5.1 Declipping.m

Po spuštění algoritmu je nejprve načten zvukový soubor. Následně je pomocí funkce `clipping` ořezán na požadovanou hodnotu prahu `tau`. Výstupní proměnnou této funkce jsou mimo ořezaného signálu i masky spolehlivých a ořezaných vzorků, které jsou prezentovány v části 4.2.

Po ořezání signálu jsou vytvořeny čtyři struktury, které obsahují všechny potřebné parametry pro segmentaci a rekonstrukci.

První ze struktur je `param`, kde jsou uloženy především základní proměnné popisující signál (počet kanálů, délka signálu, práh clippingu a masky) proměnné ovlivňující segmentaci (délka okna, posuv, typ okna) a základní parametry rekonstrukce (typ rekonstrukčního algoritmu, volba mezi analyzujícím a syntezujícím modelem, redundance transformace, maximální počet iterací a případně pořadí nízkofrekvenčního kanálu v rekonstruovaném signálu).

Další tři struktury obsahují podrobnější parametry rekonstrukčních algoritmů. Použití konkrétní z těchto 3 struktur závisí na zvoleném typu rekonstrukce v proměnné `param.rec_type`, kde je možné vybrat ze tří možností:

- `'all'` - rekonstrukce všech kanálů dohromady pomocí algoritmu CASCADE,
- `'sub'` - rekonstrukce nízkofrekvenčního kanálu zvlášť pomocí algoritmu SPADE, rekonstrukce ostatních kanálů pomocí algoritmu CASCADE,
- `'mono'` - rekonstrukce všech kanálů zvlášť pomocí funkce SPADE.

Struktura `param_multi` obsahuje proměnné pro algoritmus CASCADE, `param_sub` parametry pro rekonstrukci nízkofrekvenčního kanálu, pokud je zvolena metoda `sub` a `param_mono` parametry pro rekonstrukci pomocí algoritmu SPADE.

Dalším krokem je volání segmentační funkce, jejímž vstupem je saturovaný signál a všechny struktury. Výstupem je zrekonstruovaný signál.

Následně je vypočten a vypsán do konzole poměr  $\Delta\text{SDR}$  v jednotlivých kanálech podle vzorce (6.1c). Mimo něj je v průběhu rekonstrukce vypsán do konzole práh  $\tau$ , typ rekonstrukčního algoritmu a čas výpočtu.

## 5.2 Segmentation.m

Nejprve je po volání funkce zajištěno, aby nemohl proběhnout nesprávný typ rekonstrukce. U monofonních signálů je přiřazena do proměnné `param.rec_type` hodnota `'mono'`, takže budou zrekonstruovány pomocí algoritmu SPADE. Pokud je u stereofonního signálu nastaven typ rekonstrukce na `'sub'`, je změněn na `'all'`. Změna je vždy vypsána do konzole.

Nejprve je signál prodloužen o potřebný počet vzorků, aby všechna okna mohla mít požadovanou délku a překryv. Následně je pomocí funkce `gabwin` z toolboxu LTFAT vytvořeno analyzující okno a dvě prázdné matice o velikosti okna a počtu kanálů pro saturovaný a zrekonstruovaný signál.

Následuje cyklus `for`, ve kterém je vybrán daný segment signálu. Do struktury `param_seg` jsou předány masky pro daný blok. Následně je pomocí funkce `switch` vybrán požadovaný typ rekonstrukce, signál je v případě potřeby upraven a je volán příslušný rekonstrukční algoritmus.

Po dokončení rekonstrukce v daném bloku je blok zrekonstruovaného signálu vynásoben syntezujícím oknem a přičten k již zrekonstruovaným blokům v proměnné `data_rec_fin` a proces se opakuje pro další blok signálu.

## 5.3 Rekonstrukční algoritmy

### ASPADE.m

Funkce `ASPADE.m` provede rekonstrukci signálu podle algoritmu 1. Jejími vstupy jsou ořezaný signál, struktura `param_seg` a struktura `paramsolver` s parametry rekonstrukčního algoritmu. Výstupem je samozřejmě zrekonstruovaný blok signálu.

Po inicializaci počátečních podmínek je signál nejprve převeden z časové do frekvenční oblasti pomocí funkce `frana` toolboxu LTFAT. Následně je proveden volána funkce `hard_thresholding`, která vybere zadaný počet nevýznamnějších koeficientů.

Signál je následně převeden zpět do časové oblasti pomocí funkce `frsyn` a pomocí `projection` je provedena projekce na množinu přípustných řešení. Následně je overěna ukončovací podmínka algoritmu, pokud není splněna, jsou aktualizovány proměnné a celá operace se opakuje, dokud není podmínka splněna, nebo dokud nebylo dosaženo maximálního počtu iterací.

### SSPADE.m

Implementace syntezující verze algoritmu SPADE je velmi podobná implementaci algoritmu A-SPADE. Hlavní rozdíl tvoří především terminační kritérium a duální

proměnná  $\mathbf{u}$ , které jsou v případě algoritmu A-SPADE definovány v transformované doméně vlivem analyzujícího operátoru  $A$ , zatímco u algoritmu S-SPADE jsou vlivem operátoru  $D$  definovány v časové doméně.

### ACASCADE.m

Jelikož algoritmus CASCADE vychází z algoritmu SPADE, jsou obě implementace poměrně podobné.

Mimo rozdílu v počtu rekonstruovaných kanálů signálu spočívá hlavní rozdíl ve způsobu výběru nejvýznamnějších koeficientů. Algoritmus SPADE vybírá koeficienty pomocí funkce Hard Thresholding, algoritmus CASCADE pomocí operátoru GEW. Z toho plyne, že zatímco algoritmus SPADE vybere již v první iteraci minimálně jednu harmonickou složku, u algoritmu CASCADE může v závislosti na parametrech  $\alpha$  a především na  $\mu_0$  proběhnout relativně velké množství iterací, v nichž nebude vybrána žádná harmonická složka. Při těchto iteracích není nutné převádět signál z časové oblasti do frekvenční a zpět a provádět projekci, což by znamenalo výrazné prodloužení doby rekonstrukce.

Z toho důvodu je rekonstrukční algoritmus rozdělen na dva cykly `for`. V prvním je signál převeden do frekvenční oblasti a parametr `paramsolver.mi` je zmenšován tak dlouho, dokud není vybrán alespoň jeden koeficient.

Následně je z vybraných koeficientů ve druhém cyklu iterativně zrekonstruován poničený vstupní signál podobným postupem jako v případě algoritmu SPADE.

### SCASCADE.m

Syntezující verze algoritmu CASCADE vychází z analyzující. V implementaci jsou opět jen minimální rozdíly spojené s použitím syntezujícího modelu signálu namísto analyzujícího, podobně jako u algoritmů A-SPADE a S-SPADE.

## 5.4 Pomocné funkce

### `clipping.m`

Vstupem této funkce je původní signál a hodnota prahu. Vstupní signál je ve funkci ořezán a jsou vytvořeny masky, které jsou stěžejní pro projekci rekonstruovaného signálu na množinu přípustných řešení.

### `hard_thresholding.m`

Vstupními proměnnými jsou koeficienty  $\mathbf{c}$ , počet nejvýznamnějších koeficientů  $k$ , které mají být vybrány a struktura `param_seg`, která je zapotřebí především kvůli informaci o redundanci transformační matice.

Jelikož vektor koeficientů obsahuje vždy komplexně sdružené páry koeficientů, je vektor před výběrem  $k$ -nejvýznamnějších zkrácen na polovinu a stejnosměrná složka je vydělena

dvěma. Po výběru pomocí funkce `maxk`<sup>1</sup> je výsledný vektor opět prodloužen o komplexně sdružené koeficienty.

Funkce je implementována tak, aby bylo možné pracovat s vektorem koeficientů o sudém i lichém počtu vzorků. Z tohoto pohledu je tedy možné zvolit jakoukoliv délku rekonstrukčního okna.

### **projection.m**

Vstupními proměnnými jsou saturovaný a rekonstruovaný signál a struktura `param_seg`, obsahující masky. Na základě masek je vzorkům rekonstruovaného signálu na spolehlivých pozicích přiřazena zpět původní hodnota. Vzorkům na saturovaných pozicích je přiřazena původní hodnota, pokud je jejich současná hodnota menší (respektive v případě záporných saturovaných vzorků větší) pomocí funkce `max` (respektive `min`).

### **sdr\_func.m**

Tato funkce spočítá poměr SDR hodnoceného a referenčního signálu podle vzorce (6.1b), které jsou vstupy funkce. Funkce funguje jen pro monofonní signály, proto je nutné ji pro každý kanál vícekanálového signálu volat zvlášť.

---

<sup>1</sup>Funkce byla představena ve verzi softwaru Matlab R2017b. Pro spuštění ve starší verzi je nutné funkci nahradit.

## 6 Hodnocení výsledků algoritmu

Algoritmus CASCADE byl otestován a nastaven na 6 zkušebních nahrávkách a následně byly výsledky finálního nastavení algoritmu porovnány s výsledky algoritmu SPADE. V této kapitole nejdříve popíši nahrávky, které jsem k testování použil a následně rozeberu kritéria hodnocení nahrávek. Samotné výsledky budou následovat v další kapitole.

### 6.1 Testovací nahrávky

Tři ze šesti testovacích nahrávek použitých k zhodnocení výsledků algoritmu CASCADE byly stereofonní, zbylé tři surroundové ve formátu 5.1.

- S1 - stereo, 9 vteřin, nástrojová smyčka (flétna)
- S2 - stereo, 11 vteřin, hudební ukázka (bicí, elektrická kytara, smyčce, zpěv)
- S3 - stereo, 14 vteřin, hudební ukázka (bicí, baskytara, syntezátor)
- S4 - surround, 12 vteřin, hudební ukázka (klavír, kontrabas, flétna, zpěv)
- S5 - surround, 12 vteřin, hudební ukázka (klavír, kontrabas, flétna)
- S6 - surround, 12 vteřin, filmová hudba (smyčcové nástroje, klavír, flétna, perkusivní bicí nástroje)

Ukázka S1 je hudební smyčka z DAW softwaru Logic Pro X. Ukázky S2 až S5 jsem sám nahrál a použil se souhlasem autorů hudby. Ukázka S6 je testovací ukázka pro systémy Dolby Atmos převzatá z [31].

Vzorkovací frekvence všech nahrávek je 48 kHz a bitová slouбка 24 bitů.

### 6.2 Objektivní kritéria hodnocení

#### 6.2.1 Signal to Distortion Ratio

Poměr signal-to-distortion ratio (SDR) používají autoři algoritmu SPADE i CASCADE k hodnocení kvality rekonstruovaného signálu. Kokrétně používají hodnotu  $\Delta\text{SDR}$  podle vzorce (6.1c), což je rozdíl poměru SDR rekonstruovaného (6.1a) a saturovaného signálu (6.1b).

$$\text{SDR}_{\hat{\mathbf{x}}} = 20 \log_{10} \frac{\|[\frac{\mathbf{M}_c^+}{\mathbf{M}_c^-}]\mathbf{x}\|_2}{\|[\frac{\mathbf{M}_c^+}{\mathbf{M}_c^-}]\mathbf{x} - [\frac{\mathbf{M}_c^+}{\mathbf{M}_c^-}]\hat{\mathbf{x}}\|_2} \quad (6.1a)$$

$$\text{SDR}_{\mathbf{y}} = 20 \log_{10} \frac{\|[\frac{\mathbf{M}_c^+}{\mathbf{M}_c^-}]\mathbf{x}\|_2}{\|[\frac{\mathbf{M}_c^+}{\mathbf{M}_c^-}]\mathbf{x} - [\frac{\mathbf{M}_c^+}{\mathbf{M}_c^-}]\mathbf{y}\|_2} \quad (6.1b)$$

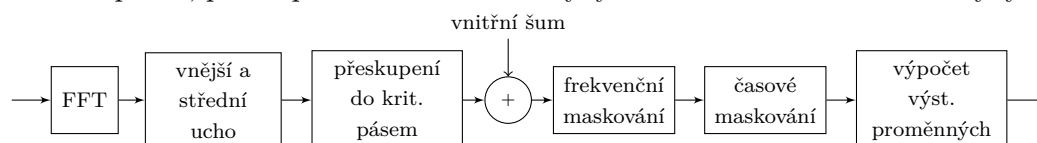
$$\Delta\text{SDR} = \text{SDR}_{\hat{\mathbf{x}}} - \text{SDR}_{\mathbf{y}} \quad (6.1c)$$

## 6.2.2 Perceptual Evaluation of Audio Quality

Metoda PEAQ je založená na hodnocení odchylky výstupů percepčního modelu pro referenční a testovaný signál, které jsou obvykle vstupem a výstupem testovaného zařízení nebo případně algoritmu [32].

Pro použití oficiální metody PEAQ podle normy ITU-R BS.1116 je třeba placená licence. Z toho důvodu je v této práci použita bezplatná verze dostupná z <http://www-mmsp.ece.mcgill.ca/Documents/Software/>.

Blokové schéma algoritmu je uvedeno na obrázku 6.1. Výstupem algoritmu je 11 proměnných MOV (*Model Output Variables*), které hodnotí změny v modulaci, hlasitost deformace, průměrnou šířku pásma, frekvence slyšitelných deformací, odstup šumu od maskovacího prahu, pravděpodobnost detekce chyby a harmonickou strukturu chyby.



Obr. 6.1: Blokové schéma základní verze percepčního modelu PEAQ.

Těchto 11 proměnných MOV je následně zpracováno kognitivním modelem, který je realizován pomocí neuronové sítě, jejíž učící množinou byly výsledky subjektivních poslechových testů. Hodnotící škála ODG (*Objective Difference Grade*) je uvedena v tabulce 6.1 a odpovídá subjektivní škále SDG. Je definována jako:

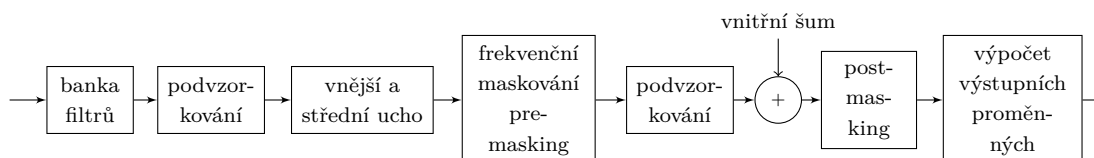
$$\text{ODG} = \text{Grade}_{\text{test}} - \text{Grade}_{\text{ref}}, \quad (6.2)$$

kde  $\text{Grade}_{\text{test}}$  je hodnocení kvality testovaného a  $\text{Grade}_{\text{ref}}$  je hodnocení kvality referenčního signálu.

Tab. 6.1: Hodnocení kvality signálu podle stupnice ODG.

Zhoršení	ODG
Nepostřehnutelné	0,0
postřehnutelné, ne nepříjemné	−1,0
mírně nepříjemné	−2,0
nepříjemné	−3,0
velmi nepříjemné	−4,0

Existuje i rozšířená verze této metody, jejíž výstupem jsou navíc MOV parametry hodnotící hlasitost deformace, změny v modulaci, hlasitost ztracených spektrálních složek a harmonické složení chyby. Blokové schéma této rozšířené verze je uvedeno na obrázku 6.2.



Obr. 6.2: Blokové schéma rozšířené verze percepčního modelu PEAQ.

## 6.3 Subjektivní kritéria hodnocení

Subjektivní hodnocení kvality signálů jsou založena na porovnávání původního a zpracovaného signálu skupinou posluchačů, kteří hodnotí kvalitu podle definované stupnice. V rámci subjektivních měření kvality signálu můžeme rozlišit několik druhů měření podle škály:

- **Nominální škála:** spočívá pouze ve třídění dat do navzájem se vylučujících kategorií na základě rovnosti nebo nerovnosti jevů
- **Pořadová škála:** data jsou seřazena na pořadové škále na základě vzájemných vztahů. Pořadová škála nemá žádnou jednotku, proto nevypovídá nic o velikosti rozdílů v rámci stupnice
- **Intervalová škála:** oproti pořadové škále je navíc možné přiřadit konkrétním jevům konkrétní hodnoty, přičemž stejně velkým subjektivním rozdílům mezi jevy odpovídají stejně velké rozdíly těchto hodnot.
- **Poměrová škála:** na poměrové škále je možné jevům přiřadit číselné hodnoty tak, aby stejně velkým poměrům velikostí mezi jevy odpovídaly stejně velké poměry mezi hodnotami stupnice.

### 6.3.1 MUSHRA

Metoda MUSHRA (*Multi Stimulus test with Hidden Reference and Anchor*) je standardizovanou metodou pro subjektivní srovnávání. Při testování dostane posluchač k posouzení několik nahrávek, mezi nimiž je známá a jedna nebo více skrytých referencí, tzv. *kotva*, což je ukázka s nejnižší kvalitou. Doporučuje se použít tzv. spodní a střední kotvu. Což je typicky referenční signál ořezaný filtrem typu dolní propust s mezní frekvencí 3,5 kHz, respektive 7 kHz. Hodnotící škála se tedy nachází mezi referencí a spodní kotvou. V této části jsem čerpal především z [32] a [33].

Jeden test může obsahovat maximálně 15 ukázek, jejichž délka nesmí přesáhnout hranici dvaceti sekund. Hodnotící škála nabývá celočíselných hodnot od 0 do 100, přičemž hodnoty jsou rozděleny do pěti skupin podle kvality (viz. tabulka 6.2).



Tab. 6.2: Hodnocení kvality signálu podle stupnice MUSHRA.

Kvalita	Hodnocení
Vynikající	100 – 80
dobrá	80 – 60
dostačující	60 – 40
špatná	40 – 20
nepříjemná	20 – 0

Absolutní výsledky nejsou příliš vhodné k prezentaci, proto se uvádí průměrná hodnota dané ukázky:

$$u_{jk} = \frac{1}{N} \sum_{i=1}^N u_{ijk}, \quad (6.3)$$

kde  $N$  je počet posluchačů a  $u_{ijk}$  je jejich absolutní hodnocení. Často jsou výsledky prezentovány jako interval, v němž s 95% pravděpodobností leží hodnota populačního průměru.

$$[u_{jk} - \delta_{jk}, u_{jk} + \delta_{jk}], \quad (6.4)$$

kde  $\delta_{jk}$  je polovina šířky intervalu spolehlivosti

$$\delta_{jk} = t_{0,05} \frac{S_{jk}}{\sqrt{N}}. \quad (6.5)$$

Proměnná  $S_{jk}$  je výběrová směrodatná odchylka a  $t_{0,05}$  je kritická hodnota Studentova rozdělení  $t$  při hladině významnosti  $\alpha = 0,05$ , která je dohledatelná ve statistických tabulkách.

$$S_{jk} = \sqrt{\sum_{i=1}^N \frac{(u_{jk} - u_{ijk})^2}{N - 1}}. \quad (6.6)$$

## 7 Výsledky

Celkem bylo v rámci testování provedeno 702 rekonstrukcí signálu pomocí algoritmu CASCADE a 288 rekonstrukcí pomocí algoritmu SPADE. V této kapitole je nejprve popsán postup nastavování algoritmu CASCADE a následně jsou zde prezentovány jeho výsledky v porovnání s algoritmem SPADE.

Pro nastavení algoritmu byly výsledky vyhodnocovány pomocí ukazatele  $\Delta\text{SDR}$ , prezentovaného v kapitole 6.2.1, což je rozdíl SDR rekonstruovaného a saturovaného signálu. Kvalita finálních výsledků byla objektivně posouzena pomocí ukazatelů  $\Delta\text{SDR}$  a PEAQ (viz. kapitola 6.2.2) a pomocí subjektivního testu MUSHRA (viz. kapitola 6.3.1).

Všechny výpočty byly provedeny v programu Matlab ve verzi R2018b na MacBooku Pro 13-inch 2016 s procesorem Intel Core i5 2 GHz s RAM LPDDR3 8 GB 1867 MHz a s operačním systémem MacOS 10.13.6 High Sierra.

### 7.1 Testy algoritmu CASCADE

#### 7.1.1 Nastavení délky okna

Nejprve byla pro algoritmus nalezena optimální délka okna pro segmentaci, při které podává nejlepší výsledky rekonstrukce podle ukazatele  $\Delta\text{SDR}$ . Nastavení ostatních parametrů ovlivňujících průběh rekonstrukce je uvedeno v tabulce 7.1.

Tab. 7.1: Výchozí nastavení parametrů rekonstrukce pomocí algoritmu CASCADE

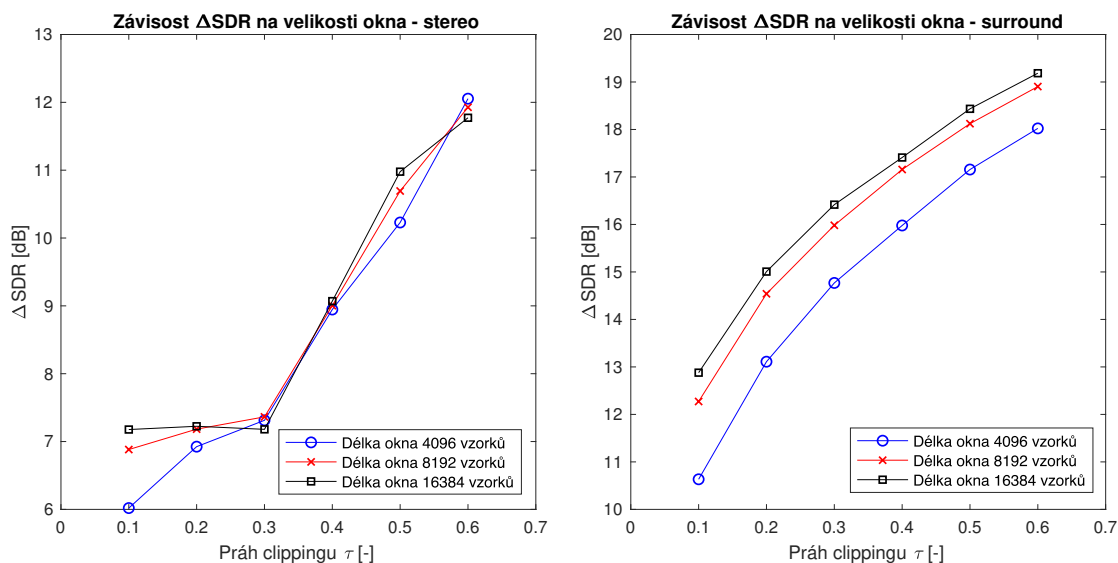
Parametr	Hodnota
Algoritmus	A-CASCADE
Překryv sousedních bloků	50 %
Redundance	1
$\alpha$	0,99
$\beta$	$10^{-3}$

Algoritmus byl testován pro délky okna 2048, 4096, 8192 a 16384 vzorků. Nejlepších výsledků bylo dosaženo s délkou okna 16384 vzorků, což při vzorkovací frekvenci 48 kHz odpovídá délce signálu 341 ms. Průměrné zlepšení  $\Delta\text{SDR}$  pro stereo signál, respektive signál ve formátu 5.1, můžeme pozorovat na grafech 7.1. Pro přehlednost není v grafech zahrnuta křivka pro délku okna 2048 vzorků. Při této hodnotě byly výsledky rekonstrukce výrazně horší, než při ostatních.

Především při rekonstrukci většího množství kanálů podává algoritmus CASCADE výrazně lepší výsledky s delším oknem, než s kratším.

Při rekonstrukci stereofonního signálu lze výhody delšího okna pozorovat především na velmi saturovaných signálech. Na méně poškozených signálech se rozdíly mezi různými velikostmi okna stírají.

Další nespornou výhodou dlouhého okna je také výrazně kratší čas výpočtu, především v případě více saturovaných signálů. U stereofonních signálů proběhla rekonstrukce při délce okna 16384 vzorků a prahu  $\tau = 0,1$  v průměru o 25 % rychleji než při poloviční délce okna, u vícekanálových signálů o 18 %. S rostoucím prahem, a tím pádem s klesajícím počtem saturovaných vzorků, se rozdíl v času rekonstrukce postupně snižoval. Při hodnotě prahu  $\tau = 0,6$  pracoval algoritmus stejně rychle při obou zmiňovaných délkách okna při rekonstrukci stereofonních signálů. Při rekonstrukci surroundových signálů to bylo dokonce již od  $\tau = 0,3$ .



Obr. 7.1: Závislost kvality rekonstrukce pomocí algoritmu A-CASCADE na délce okna.

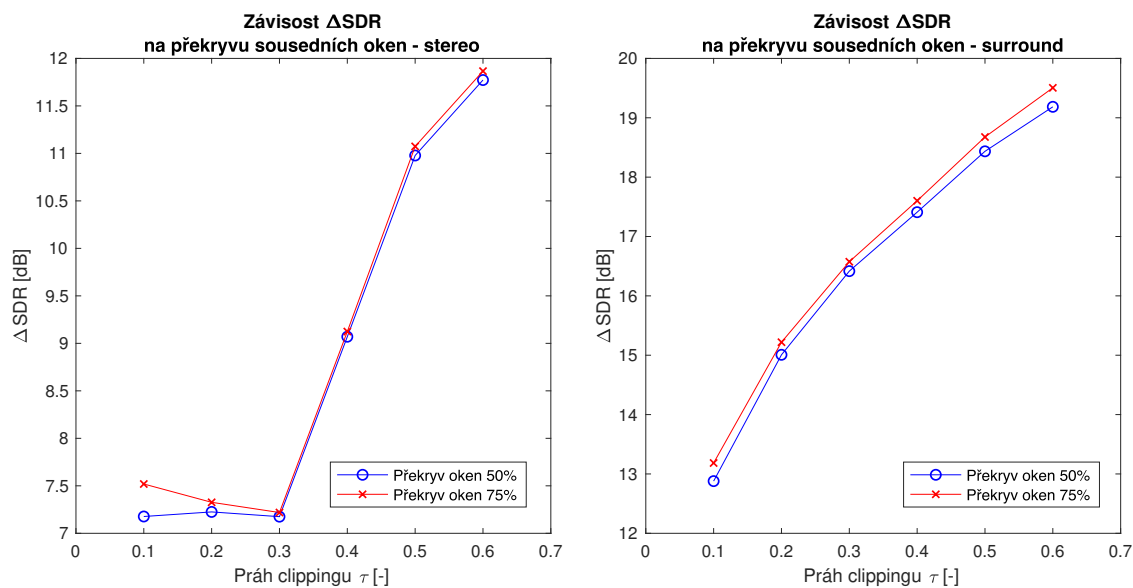
### 7.1.2 Nastavení posuvu okna

Dalším nastavovaným parametrem byl posuv okna. Algoritmus byl testován pro 50% a 75% překryv. Výsledky pro testovací signály jsou uvedeny na grafech 7.2.

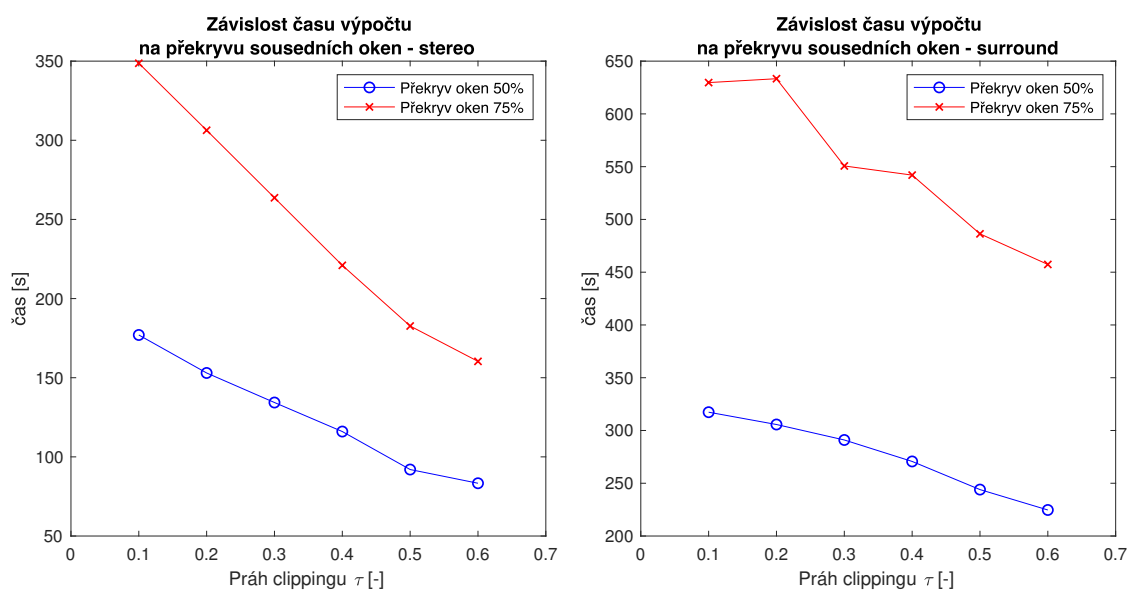
Lepší výsledky byly dosaženy při překryvu 75 % (v průměru asi o 0,3 dB větší poměr  $\Delta SDR$  než u 50% překryvu u vícekanálových signálů). U stereofonních signálů se rozdíl projevoval především na více saturovaných signálech.

Nevýhodou, která plyne z většího množství oken, je samozřejmě delší čas výpočtu. Je otázkou, zda je v tomto případě zlepšení skutečně tak markantní, aby bylo prodloužení doby výpočtu při použití čtvrtinového posuvu okna přínosné.

Z grafů 7.3 je patrné, že rekonstrukce při 75 % překryvu trvá přibližně dvakrát déle. Z toho důvodu jsem se rozhodl ostatní testy, které jsou poměrně časově náročné, provést pouze s polovičním posuvem oken a čtvrtinový posuv aplikovat až při závěrečných porovnáních algoritmu CASCADE s algoritmem SPADE.



Obr. 7.2: Závislost kvality rekonstrukce pomocí algoritmu A-CASCADE na velikosti překryvu oken.



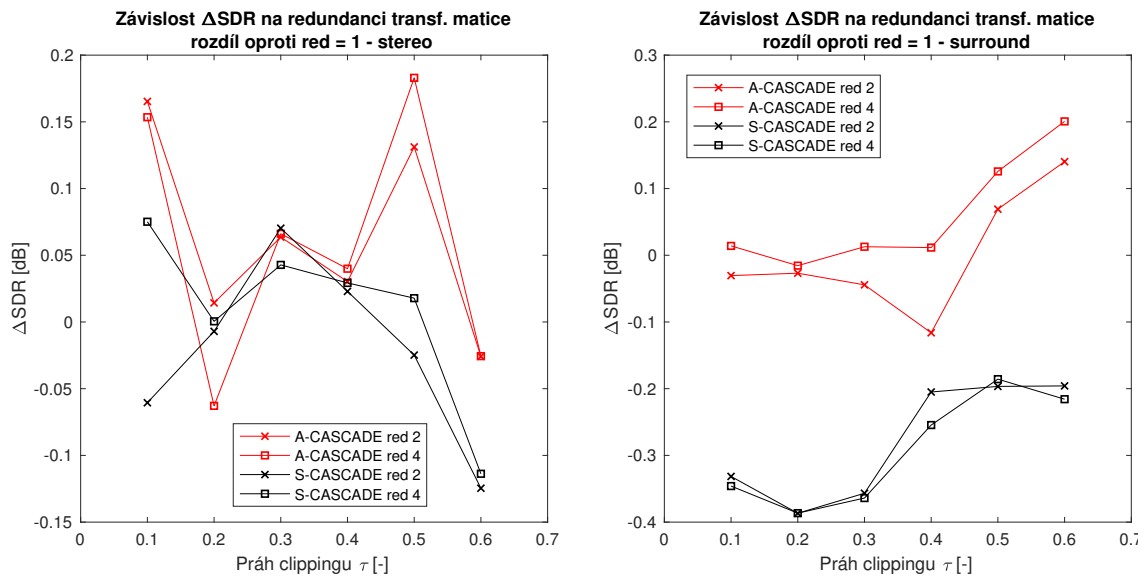
Obr. 7.3: Závislost rychlosti rekonstrukce pomocí algoritmu A-CASCADE na velikosti překryvu oken.

### 7.1.3 Nastavení redundance

Dalším krokem při nastavování algoritmu CASCADE byl výběr vhodné transformační matice a verze algoritmu. Při redundanci 1 je počet koeficientů po transformaci signálu do kmitočtové oblasti roven počtu vzorků daného signálu (v tomto případě délce okna, tj. 16384). Při redundanci 2 je počet koeficientů zdvojnásoben. V praxi to znamená, že je možné řídkou reprezentaci daného signálu sestavit z většího množství lineárních kombinací sloupců transformační matice, což může vést k lepším výsledkům.

Testy byly provedeny pro analyzující a syntezující verzi algoritmu CASCADE při redundanci transformace 2 a 4. Test algoritmu S-CASCADE pro redundanci 1 nebylo nutné provádět, protože z principu podává stejné výsledky jako A-CASCADE při stejném nastavení (viz. podkapitola 3.3).

Rozdíly mezi rekonstruovanými signály při různých hodnotách redundancí nebyly příliš patrné, proto je na grafech 7.4 uveden rozdíl  $\Delta\text{SDR}$  pro danou hodnotu redundance oproti  $\Delta\text{SDR}$  při redundanci 1.

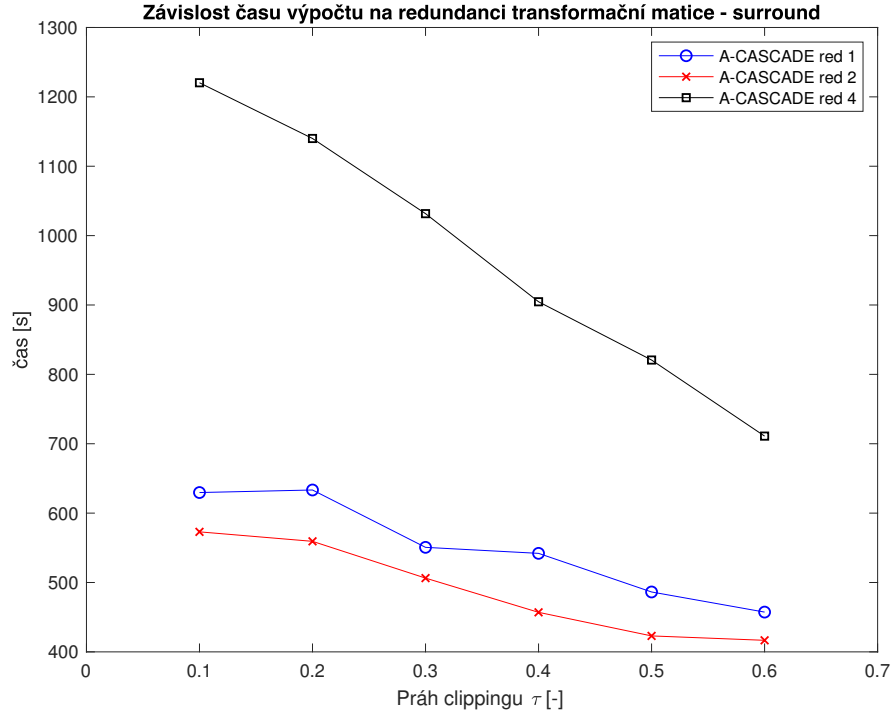


Obr. 7.4: Závislost kvality rekonstrukce pomocí algoritmu CASCADE na redundanci transformace.

Z grafu, kde je zobrazen rozdíl hodnot  $\Delta\text{SDR}$  pro stereofonní signály, lze pozorovat, že použití redundantní transformační matice nemusí být vždy přínosem. Například transformační matice s redundancí 4 může v některých případech podávat mnohem horší výsledky, navíc při mnohem delší době výpočtu.

Z grafu je patrné, že použití redundantní matice je značně nevyzpytatelné. Pouze algoritmus A-CASCADE s redundancí 2 podává mírně lepší výsledky než při redundanci 1.

Zatímco u stereofonních signálů podávala syntezující verze algoritmu téměř srovnatelné výsledky s analyzující verzí, při rekonstrukci vícekanálových signálů je mezi oběma verzemi patrný značný rozdíl. Pro vícekanálové signály se jeví jako nejlepší možnost transformační matice s redundancí 4. Při tomto nastavení však doba výpočtu vzroste na téměř dvojnásobek oproti redundanci 2 nebo 1 (viz. graf 7.5). Z toho důvodu jsem se rozhodl v dalších testech používat algoritmus A-CASCADE s dvojnásobným počtem koeficientů, stejně jako v případě stereofonních signálů.



Obr. 7.5: Závislost rychlosti rekonstrukce vícekanálového signálu pomocí algoritmu A-CASCADE na redundanci transformační matice

#### 7.1.4 Nastavení parametru $\alpha$

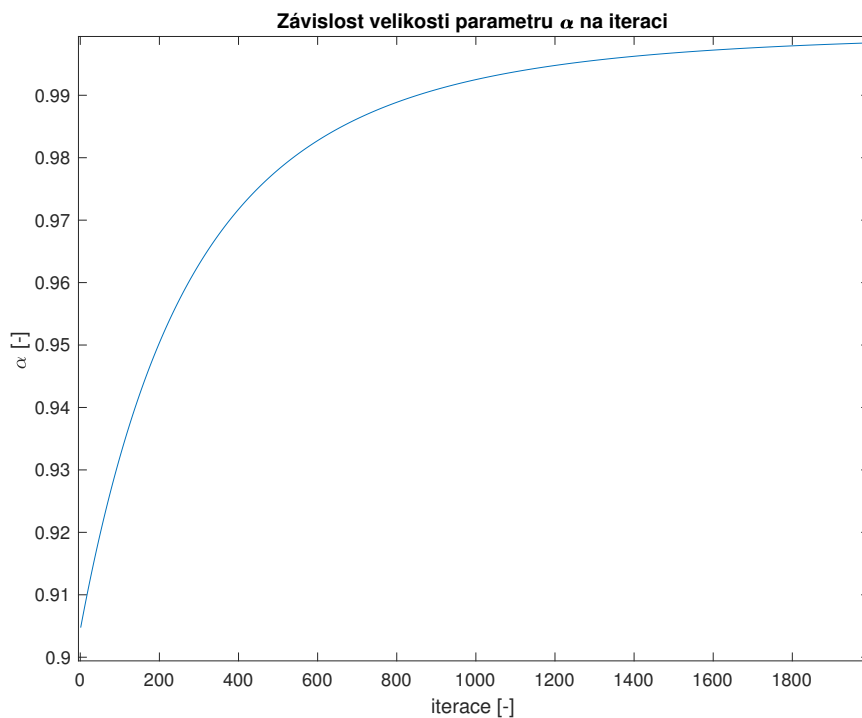
Dalším testovaným parametrem byl parametr  $\alpha$ , který ovlivňuje „jemnost“ při výběru koeficientů v průběhu iterací algoritmu CASCADE. Algoritmus A-CASCADE byl postupně otestován pro  $\alpha \in \{0,9; 0,93; 0,95; 0,97; 0,99\}$  a pro progresivní parametr  $\alpha$  podle rovnice:

$$\alpha = 1 - \left( \frac{1}{1,8 + 0,0016i} \right)^4, \quad (7.1)$$

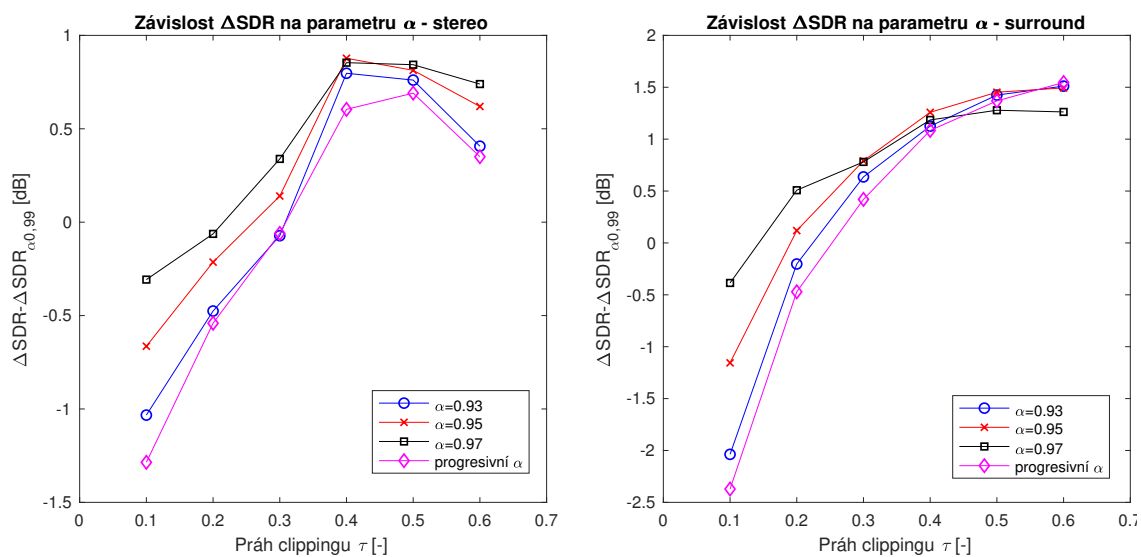
kde  $i$  je číslo dané iterace. Průběh této funkce je zaznamenán na grafu 7.6.

Na grafech 7.7 je vyneseno rozdílné hodnoty  $\Delta\text{SDR}$  pro jednotlivé hodnoty parametru  $\alpha$  oproti výchozí hodnotě  $\Delta\text{SDR}_{\alpha 0,99}$ . Pro větší přehlednost jsme vynechali křivku pro  $\alpha = 0,9$ . Při této hodnotě podával algoritmus jasně nejhorší výsledky.

Z testovaných hodnot byla nakonec do dalších testů vybrána  $\alpha = 0,97$ , která sice podává mírně horší výsledky při rekonstrukci výrazně poškozených signálů, ale při rekonstrukci surroundových signálů ořezaných prahem  $\tau \geq 0,3$  dochází ke zlepšení průměrné hodnoty  $\Delta\text{SDR}$  přibližně o 1 dB.



Obr. 7.6: Průběh progresivního parametru  $\alpha$ .



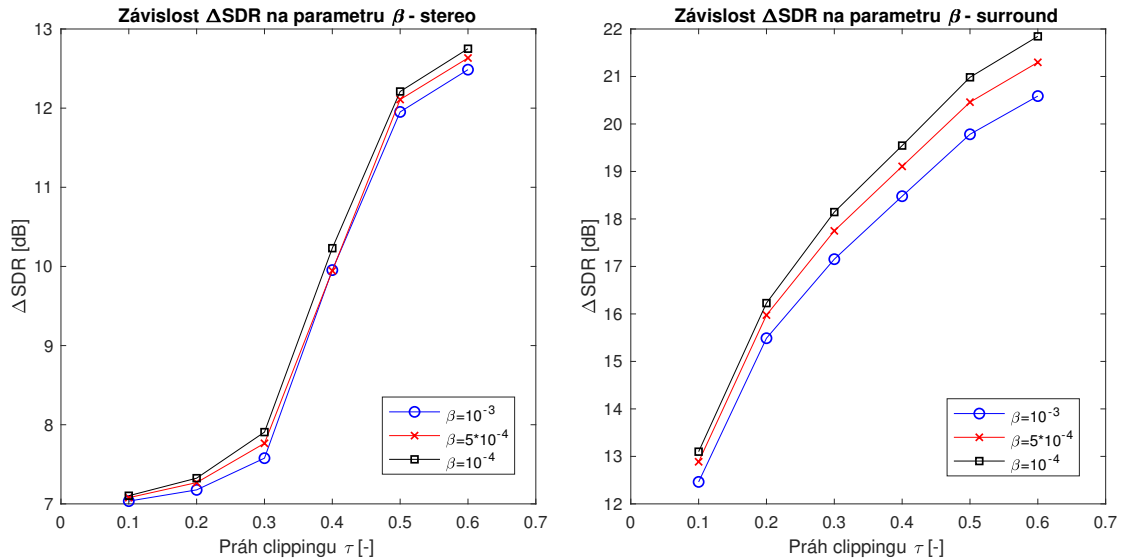
Obr. 7.7: Závislost kvality rekonstrukce pomocí algoritmu A-CASCADE na hodnotě parametru  $\alpha$ .

### 7.1.5 Nastavení parametru $\beta$

Podmínka na řádce 4, v algoritmu 3, která rozhoduje o ukončení rekonstrukčního algoritmu v každém bloku byla mírně upravena. Parametr  $\beta$  je při rozhodování vynásoben počtem kanálů, aby bylo zajištěno, že průměrný rozdíl v každém kanálu mezi signálem

před a po rekonstrukci bude odpovídat hodnotě  $\beta$ . Díky této drobné úpravě není nutné nastavovat tento parametr rozdílně pro signál s různým počtem kanálů.

Test byl proveden pro 3 hodnoty  $\beta \in \{10^{-3}; 5 \cdot 10^{-4}; 10^{-4}\}$ , jejichž výsledky jsou uvedeny na grafech 7.8.



Obr. 7.8: Závislost kvality rekonstrukce pomocí algoritmu A-CASCADE na hodnotě parametru  $\beta$ .

### 7.1.6 Rekonstrukce korelovaných kanálů signálu zvlášť

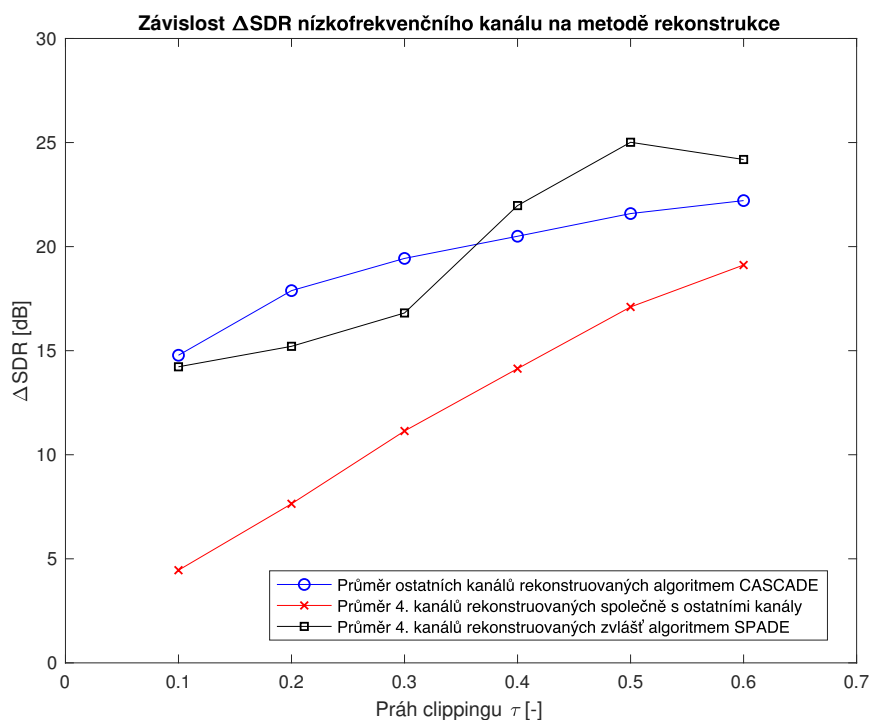
#### Rekonstrukce nízkofrekvenčního kanálu

V průběhu testování surroundových nahrávek byl pozorován nežádoucí jev, kdy u velmi poškozených signálů docházelo pouze k malému zlepšení odstupu SDR ve čtvrtém, tedy efektovém, kanálu. Průměrný rozdíl oproti ostatním kanálům při  $\tau = 0,1$  činil víc než 10 dB. Při rostoucím  $\tau$  se rozdíl zmenšoval, ale  $\Delta\text{SDR}$  nízkofrekvenčního kanálu stále nedosahoval průměrných hodnot ostatních kanálů.

Řešením bylo čtvrtý kanál oddělit od ostatních kanálů a rekonstruovat jej zvlášť pomocí algoritmu SPADE. Tento krok přinesl očekávané zlepšení, avšak po oddělení nízkofrekvenčního kanálu došlo ke zhoršení průměrného  $\Delta\text{SDR}$  u ostatních kanálů. Jako nejlepší možnost se tedy jeví rekonstrukce všech kanálů dohromady pomocí algoritmu CASCADE a až poté zrekonstruovat čtvrtý kanál samostatně pomocí algoritmu SPADE.

Tento krok samozřejmě vede k prodloužení doby výpočtu, v průměru téměř o třetinu. Na druhou stranu zlepšení  $\Delta\text{SDR}$  se u rekonstrukce velmi poškozených signálů pohybuje až okolo 10 dB. Proto jsem se rozhodl tuto metodu využít i při finálním srovnání výsledků algoritmů SPADE a CASCADE. K rekonstrukci nízkofrekvenčního kanálu bylo použito nastavení algoritmu SPADE uvedené v tabulce 7.2.





Obr. 7.9: Závislost kvality rekonstrukce nízkofrekvenčního kanálu na metodě rekonstrukce

Tab. 7.2: Nastavení algoritmu SPADE pro rekonstrukci nízkofrekvenčního kanálu.

Parametr	Hodnota
Redundance	2
$\varepsilon$	0,1
$s$	1
$r$	1

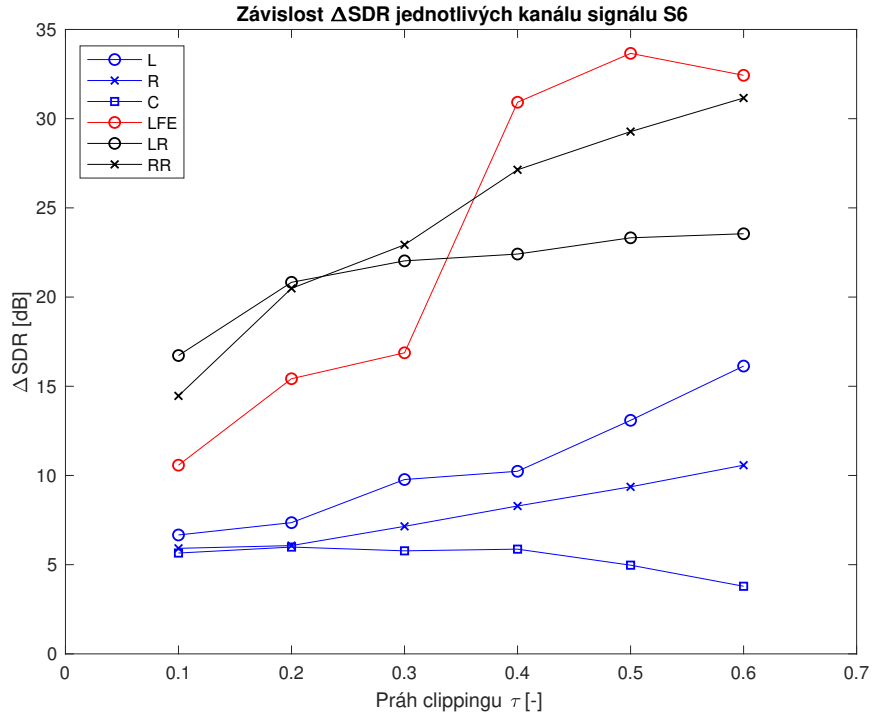
### Rekonstrukce dvou skupin korelovaných kanálů

K dalším problémům docházelo u signálu S6, kde nejsou kanály příliš korelovány. Průběhy  $\Delta\text{SDR}^1$  v jednotlivých kanálech můžeme pozorovat na grafu 7.10.

Řešením tohoto problému by mohlo být rozdělit signál na dvě (eventuelně na více) skupiny kanálů a každou skupinu rekonstruovat zvlášť.

Tato možnost byla experimentálně ověřena. Nejprve bylo v každém bloku ověřena míra korelace dvou nejvíce rozdílných kanálů. Pokud byly oba signály dostatečně korelovány, byl celý blok zrekonstruován dohromady, pokud ne, tyto dva kanály se staly hlavními prvky dvou skupin. Poté byl do každé skupiny přiřazen signál dalšího kanálu, který byl nejvíce podobný některému ze signálů ve skupině.

<sup>1</sup>Čtvrtý kanál je rekonstruovaný zvlášť pomocí metody SPADE



Obr. 7.10: Hodnota  $\Delta\text{SDR}$  jednotlivých kanálů signálu S6.

Během tohoto pokusu však ke žádnému zlepšení  $\Delta\text{SDR}$  v prvních třech kanálech signálu S6 nedošlo, naopak ve všech kanálech došlo k mírnému zhoršení. To vše navíc při prodloužení doby výpočtu. S ohledem na citlivost rekonstrukčních algoritmů na změny v nastavení je však stále možné, že by podobný postup mohl vést k úspěšnému cíli a je zde tedy prostor k dalšímu výzkumu v oblasti rekonstrukce vícekanálových signálů pomocí algoritmů CASCADE a SPADE.

## 7.2 Nastavení parametrů algoritmu SPADE

Aby porovnání algoritmů SPADE a CASCADE bylo relevantní, bylo potřeba optimálně nastavit i algoritmus SPADE. Při jeho nastavení jsem sice vyšel z článku [34], ale ten se zabývá rekonstrukcí signálu se vzorkovací frekvencí 16 kHz, zatímco v této práci všechny rekonstrukce probíhají při vzorkovací frekvenci 48 kHz. Při změně vzorkovací frekvence může být chování rekonstrukčního algoritmu dosti odlišné.

Tento test probíhal naprosto shodně s testem algoritmu CASCADE pouze s tím rozdílem, že byl vždy rekonstruován jen první kanál každého signálu. Ideální délka okna a překryv oken je stejný jako u algoritmu CASCADE - 16384 vzorků a 75 %. Co se redundance transformační matice týče, podává lepší výsledky také dvojnásobná, rozdíl je však více patrný, než u algoritmu CASCADE.

Byl proveden i test parametru  $s$ , který určuje, kolik harmonických složek bude v každé iteraci vybráno. Článek [16] i [34] uvádí, že optimální hodnota je  $s = 1$ , v testech se však projevila jako výrazně lepší pro méně saturované signály hodnota  $s = 3$ , kdy byl rozdíl

v  $\Delta\text{SDR}$  v řádu jednotek decibelů. Naopak při nižších hodnotách prahu podával algoritmus s tímto nastavením mírně horší výsledky, zhoršení však bylo v řádu setin decibelu.

Toto nastavení se však ukázalo jako nevhodné pro rekonstrukci nízkofrekvenčního kanálu surroundových signálů. Pro tuto rekonstrukci je tedy vhodně použít nastavení uvedené v tabulce 7.2

## 7.3 Porovnání algoritmů SPADE a CASCADE

Jako poslední byl proveden srovnávací test obou algoritmů. Nastavení obou je uvedeno v tabulce 7.3.

Tab. 7.3: Nastavení algoritmů SPADE a CASCADE pro srovnávací test.

CASCADE		SPADE	
Délka okna	16384	Délka okna	16384
Překryv oken	75 %	Překryv oken	75 %
Redundance	2	Redundance	2
$\alpha$	0,97	$s$	3
$\beta$	$10^{-4}$	$\varepsilon$	0,3

Test probíhal v podstatě velmi podobně jako průběžné testy během nastavování algoritmu. Každý signál byl ořezán šesti prahy  $\tau \in \{0,1; 0,2; 0,3; 0,4; 0,5; 0,6\}$ . Každý takto ořezaný signál byl následně zrekonstruován pomocí algoritmů A-SPADE, S-SPADE, A-CASCADE a S-CASCADE<sup>2</sup>.

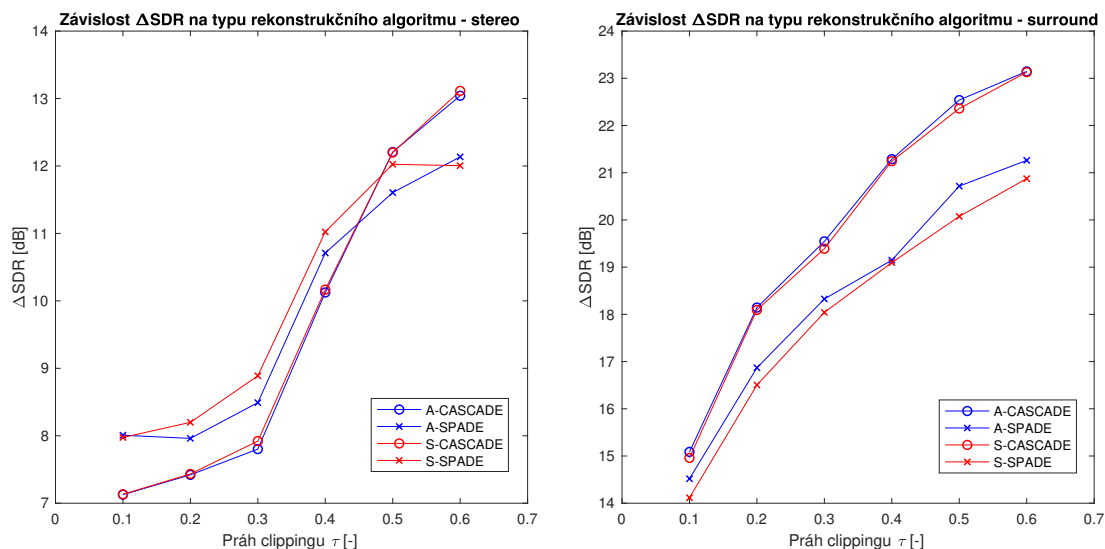
Výsledká kvalita zrekonstruovaných signálů byla následně posouzena pomocí poměru  $\Delta\text{SDR}$ , metody PEAQ a pomocí subjektivního poslechového testu MUSHRA.

V této kapitole budou prezentovány pouze průměrné výsledky pro stereofonní a vícekanálové signály. Grafy s podrobnějšími výsledky v jednotlivých kanálech jednotlivých ukázek jsou součástí přílohy (příloha A.1 obsahuje grafy s hodnotami  $\Delta\text{SDR}$  a příloha A.2 s hodnotami ODG podle ukazatele PEAQ) a tabulka s konkrétními hodnotami je součástí elektronické verze této práce. V příloze A.3 jsou uvedena průměrná hodnocení jednotlivých signálů podle subjektivního poslechového testu MUSHRA včetně intervalu, v němž s 95% pravděpodobností leží hodnota populačního průměru.

### 7.3.1 Výsledky SDR

Z grafů 7.11 můžeme pozorovat, že podle ukazatele  $\Delta\text{SDR}$  algoritmy SPADE fungují lépe pro stereofonní signály, zatímco CASCADE podávají lepší výsledky při rekonstrukci signálů s větším množstvím kanálů.

<sup>2</sup>Při rekonstrukci pomocí algoritmu CASCADE byla použita verze algoritmu, při které je nízkofrekvenční kanál zrekonstruován zvlášť pomocí algoritmu SPADE.



Obr. 7.11: Závislost kvality rekonstrukce na typu rekonstrukčního algoritmu.

Důležitou roli však při volbě algoritmu může hrát i doba výpočtu. Ta je u algoritmu CASCADE výrazně nižší. Průměrná doba rekonstrukce pro jednotlivé algoritmy je uvedena v tabulce 7.4

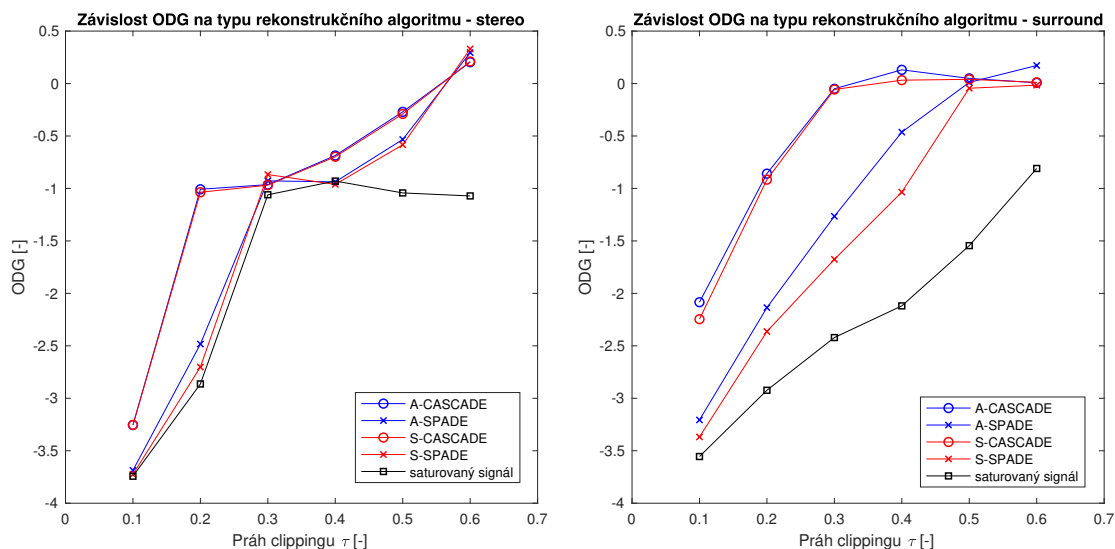
Tab. 7.4: Porovnání efektivity algoritmů SPADE a CASCADE.

-	Stereo		Surround 5.1	
	prům. čas	prům. $\Delta\text{SDR}$	prům. čas	prům. $\Delta\text{SDR}$
A-SPADE	989 s	9,818 dB	1110 s	18,473 dB
S-SPADE	875 s	10,019 dB	879 s	18,118 dB
A-CASCADE	156 s	9,620 dB	536 s	19,958 dB
S-CASCADE	121 s	9,661 dB	333 s	19,863 dB

Z tohoto pohledu je nejefektivnější syntezující verze algoritmu CASCADE. Pro stereofonní signály lze dosáhnout lepších výsledků s algoritmem S-SPADE, však za cenu téměř sedminásobné doby výpočtu. U vícekanálových signálů podávají algoritmy A-CASCADE a S-CASCADE téměř stejné výsledky, avšak syntezující verze ve výrazně kratším čase.

### 7.3.2 Výsledky PEAQ

Podle výsledků metody PEAQ je kvalita rekonstruovaných signálů výrazně lepší po rekonstrukci metodou CASCADE a to i u stereofonních signálů, přičemž výsledky analyzující a syntezující verze jsou prakticky totožné. Navíc vícekanálové signály poničené prahem  $\tau = 0,3$  a vyšším a rekonstruované metodou CASCADE jsou podle ukazatele ODG nerozaznatelné od původního signálu.



Obr. 7.12: Závislost kvality rekonstrukce signálu na typu rekonstrukčního algoritmu.

Při hodnocení metodou PEAQ však bylo zjištěno, že kvalita signálu nízkofrekvenčního kanálu je po rekonstrukci algoritmem SPADE tristní, pokud bylo poškození signálu před rekonstrukcí značné.

Na základě toho bylo provedeno důkladnější porovnání kvality rekonstrukce pomocí metody PEAQ nízkofrekvenčních signálů rekonstruovaných oběma metodami. Jak lze pozorovat na grafu<sup>3</sup> 7.13, ODG signálů rekonstruovaných metodou CASCADE je u velmi poškozených signálů mnohem lepší, než u signálů rekonstruovaných metodou SPADE, přestože podle ukazatele  $\Delta\text{SDR}$  by tomu mělo být opačně.

Ukazatel  $\Delta\text{SDR}$  je sice nejrychlejším způsobem, jak vyhodnotit kvalitu nahrávek, ale není nejspolehlivějším. Nahrávky nízkofrekvenčního kanálu bohužel nebyly zahrnuty do subjektivního poslechového testu, ale jejich kvalita byla poslechem alespoň ověřena.

U nízkofrekvenčních signálů s velkou mírou zkreslení je po rekonstrukci zkreslení stále slyšitelné ve všech případech, ale u signálů rekonstruovaných pomocí algoritmů CASCADE je mnohem méně rušivé. Z toho usuzuji, že výsledky hodnocení metodou PEAQ odpovídají realitě více, než výsledky poměru  $\Delta\text{SDR}$ .

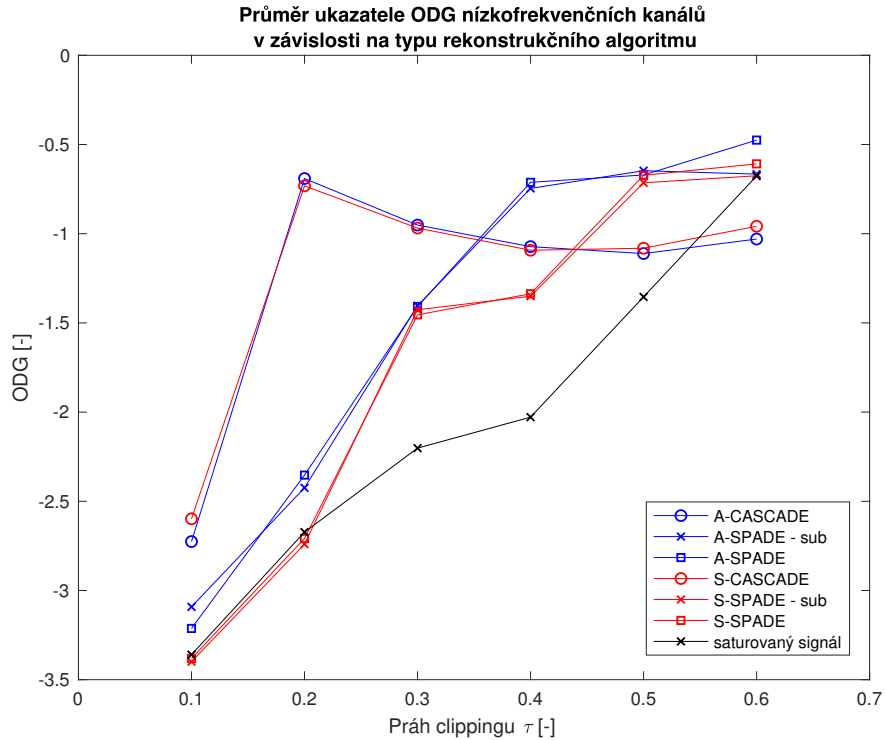
Většina zkreslení, které signál obsahuje i po rekonstrukci je relativně vysoko nad mezní frekvencí tohoto kanálu, která je 120 Hz. Tudíž se nabízí možnost, toto zkreslení odstranit pomocí dostatečně strmého filtru typu dolní propust.

Nicméně zkreslení se může vyskytovat i pod touto hranicí, proto by bylo vhodné pro rekonstrukci nízkofrekvenčních signálů použít jiný rekonstrukční algoritmus.

K tomuto závěru jsem bohužel došel poměrně pozdě a k hledání optimálního rekonstrukčního algoritmu pro nízkofrekvenční kanály vícekanálových audiosignálů mi nezbylo dostatek času. Domnívám se však, že upravený algoritmus SPADE nebo

<sup>3</sup>Poznámka „sub“ u algoritmu SPADE znamená, že bylo použito nastavení uvedené v tabulce 7.2, které je podle ukazatele  $\Delta\text{SDR}$  vhodnější pro rekonstrukci nízkofrekvenčních signálů.

CASCADE, který by při rekonstrukci přisuzoval větší váhu koeficientům v nízkofrekvenční části spektra by pro tento účel mohl fungovat poměrně dobře.



Obr. 7.13: Závislost kvality rekonstrukce nízkofrekvenčního kanálu na metodě rekonstrukce.

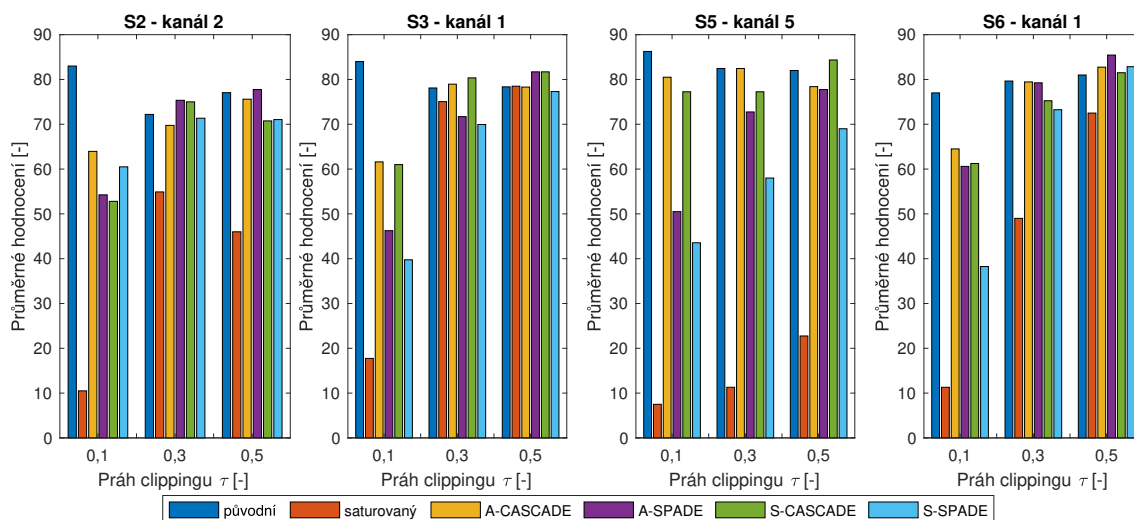
### 7.3.3 Výsledky MUSHRA

Kvalita rekonstruovaného signálu byla ověřena také pomocí subjektivního poslechového testu MUSHRA. Test obsahoval celkem 72 nahrávek, z toho bylo 12 referenčních, tedy neoříznutých, původních. Dalších 12 byly kotvy a zbývajících 48 ukázek byly rekonstruované.

Ukázky byly rozděleny do 12 bloků, každý obsahoval originální nahrávku, saturevanou nerekonstruovanou a čtyři rekonstruované ukázky pomocí analyzujících a syntezujících verzí algoritmů SPADE a CASCADE. Jako ukázka pro subjektivní poslechový test byl vybrán vždy jeden kanál z kompletní rekonstruované vícekanálové nahrávky. Konkrétně se jednalo o signály:

- S2 – kanál 2
- S3 – kanál 1
- S5 – kanál 5
- S6 – kanál 1

Každý z těchto signálů byl obsažen ve třech blocích, pokaždé z jinou prahovou úrovní clippingu ( $\tau \in \{0,1; 0,3; 0,5\}$ ).



Obr. 7.14: Průměrné hodnocení kvality nahrávek podle poslechového testu MUSHRA.

Z výsledků subjektivního poslechového testu uvedených v grafu 7.14 a kompletních výsledků v příloze A.3 lze usuzovat, že i relativně hodně poškozený signál nemusí být pro posluchače v některých případech příliš rušivý (například signál S3 při  $\tau = 0,3$ ).

Zajímavé je hodnocení původního signálu, který byl vždy pro trojici bloků stejný, dosahoval však různých hodnocení. Ve většině případů dosahoval původní signál vyššího průměrného hodnocení, pokud byli posluchači schopni jasně identifikovat kotvu.

Posluchači ve většině případů neměli problém s rozpoznáním referenčního signálu, jehož průměrné hodnocení se pohybovalo kolem 80 bodů. Ve většině případů posluchači rovněž jasně identifikovali kotvu, jediný signál u kterého byla kotva problematicky rozpoznatelná byl signál S3 při prahu clippingu  $\tau = 0,3$  a  $\tau = 0,5$ .

Z rekonstrukčních algoritmů podává podle MUSHRA testu nejlepší výsledky algoritmus A-CASCADE, který i při rekonstrukci velmi saturevaných signálů hodnotí posluchači jako „dobrý“ (v průměru nejméně 61,6 bodů).

Ve většině případů podává srovnatelné, někdy i mírně lepší, výsledky i algoritmus S-CASCADE. V případě signálu S2 při prahu clippingu  $\tau = 0,1$  je však hodnocen jako výrazně horší (přibližně o 11 bodů).

Algoritmus SPADE při rekonstrukci signálů, u kterých je poškození jasně patrné, podle výsledků poslechových testů ve většině případů výrazně zaostává. V některých případech však může podávat mírně lepší výsledky než algoritmus CASCADE.

Z poslechových testů tedy lze vyvodit závěr, že při použití rekonstrukčního algoritmu A-CASCADE nemusí být vždy dosaženo nejlepšího výsledku, na druhou stranu nikdy nebyl v průběhu testů hodnocen hůře než „dobrý“. Z toho usuzují, že použití tohoto algoritmu je nejjistější variantou pro kvalitní rekonstrukci saturevaného vícekanálového signálu.

## 8 Závěr

V této diplomové práci jsem se věnoval problematice rekonstrukce vícekanálového saturevaného signálu.

V první části práce se věnuji teoretickým základům, potřebným k implementaci rekonstrukčního algoritmu. Druhá část se věnuje implementaci těchto algoritmů v prostředí Matlab a ve třetí části jsou zhodnoceny výsledky rekonstrukce.

V první kapitole je podrobně rozebrána problematika clippingu a jsou zde popsány jeho důsledky. Ve druhé kapitole jsou krátce shrnuty dosavadní algoritmy, pomocí kterých je možné ořezaný signál rekonstruovat. Nejlepších výsledků dosahují metody založené na řídké reprezentaci signálů. Za „state-of-the-art“ pro rekonstrukci monofonních saturevaných signálů je považován algoritmus SPADE, který je schopen rekonstruovat poškozený signál velmi efektivně v relativně krátkém čase.

Třetí kapitola tvoří teoretický základ pro pochopení problematiky řídké reprezentace signálu. Jsou zde popsány báze a framy, které jsou používány při rekonstrukci jako transformační matice, strukturovaná řídkost, které se využívá při rekonstrukci pomocí algoritmu CASCADE a hledání řídkého řešení pomocí metody ADMM.

Čtvrtá kapitola je věnována hledání řídkého řešení v praxi. Je zde popsána problematika segmentace a postup při hledání řídkého řešení pomocí algoritmů SPADE a CASCADE. Jsou zde rozebrány oba algoritmy při použití jak analyzující, tak syntezující transformace signálu. V páté kapitole je detailně popsána implementace obou algoritmů v prostředí Matlab, včetně všech pomocných funkcí, jako je hard thresholding, segmentace nebo projekce.

V další kapitole se teoreticky věnuji hodnocení kvality rekonstruovaného signálu. Nejprve jsou zde popsány testovací signály, pomocí kterých byly rekonstrukční algoritmy nejprve nastaveny a následně porovnány. Poté jsou zde popsány metody hodnocení a sice ukazatele  $\Delta\text{SDR}$  a  $\text{PEAQ}$  jako objektivní ukazatele a poslechový test MUSHRA jako subjektivní ukazatel kvality.

Sedmá kapitola je kapitolou stěžejní a věnuje se nejprve správnému nastavení obou algoritmů a následně jejich srovnání pro rekonstrukci vícekanálových saturevaných audio signálů. V této kapitole jsou popsány i snahy najít vhodný způsob, jak v rámci rekonstrukce signálu pomocí algoritmu CASCADE provádět rekonstrukci po skupinách korelovaných kanálů. Tyto pokusy však nepřinesly mnoho úspěchu a to ani v případě samostatné rekonstrukce nízkofrekvenčního kanálu signálu ve formátu 5.1 Surround, kdy sice výsledky  $\Delta\text{SDR}$  byly lepší než při společné rekonstrukci, ale při hodnocení  $\text{PEAQ}$  se ukázaly jako mnohem méně zkreslené nízkofrekvenční signály rekonstruované společně s ostatními kanály algoritmem CASCADE.



Z výsledků uvedených v přílohách lze pozorovat, že nelze jednoznačně určit, který algoritmus bude pro daný signál podávat nejlepší výsledky, neboť záleží na mnoha faktorech. Z trochou zobecnění lze ale říci, že lepší výsledky podávají častěji algoritmy CASCADE, především při větším množství rekonstruovaných kanálů. Tyto algoritmy navíc pracují mnohem efektivněji.

## Literatura

- [1] DEFRAENE, Bruno, et al. Real-time perception-based clipping of audio signals using convex optimization. *IEEE Transactions on Audio, Speech, and Language Processing*, 2012, 20.10: 2657-2671.
- [2] Wikipedie: Otevřená encyklopedie: Clipping (zvuk) [online]. c2017 [citováno 8. 12. 2018]. Dostupný z WWW: <[https://cs.wikipedia.org/w/index.php?title=Clipping\\_\(zvuk\)&oldid=15125900](https://cs.wikipedia.org/w/index.php?title=Clipping_(zvuk)&oldid=15125900)>
- [3] HARVILLA, Mark J.; STERN, Richard M. Efficient audio declipping using regularized least squares. *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE*, 2015. p. 221-225.
- [4] ESQUEDA, Fabián; BILBAO, Stefan; VÄLIMÄKI, Vesa. Aliasing reduction in clipped signals. *IEEE Transactions on Signal Processing*, 2016, 64.20: 5255-5267.
- [5] KOVAČEVIĆ, Svjetlana; PEŠIĆ-BRĐANIN, Tatjana; GALIĆ, Jovan. Intermodulation distortion of class D audio amplifier using pulse density modulation. *Zooming Innovation in Consumer Electronics International Conference (ZINC)*, 2016. IEEE, 2016. p. 46-49.
- [6] GODSILL, Simon J.; WOLFE, Patrick J.; FONG, William NW. Statistical model-based approaches to audio restoration and analysis. *Journal of New Music Research*, 2001, 30.4: 323-338.
- [7] FONG, William; GODSILL, Simon. Monte Carlo smoothing for non-linearly distorted signals. *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on. IEEE*, 2001. p. 3997-4000.
- [8] ABEL, Jonathan S.; SMITH, Julius O. Restoring a clipped signal. *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on. IEEE*, 1991. p. 1745-1748.
- [9] DAHIMENE, Abdelhakim; NOUREDDINE, Mohamed; AZRAR, Arab. A simple algorithm for the restoration of clipped speech signal. *Informatica*, 2008, 32.2.
- [10] ADLER, Amir, et al. A constrained matching pursuit approach to audio declipping. *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE*, 2011. p. 329-332.
- [11] WEINSTEIN, Alejandro J.; WAKIN, Michael B. Recovering a clipped signal in sparseland. *arXiv preprint arXiv:1110.5063*, 2011.
- [12] KITIĆ, Srđan, et al. Consistent iterative hard thresholding for signal declipping. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE*, 2013. p. 5939-5943.

- [13] DEFRAENE, Bruno, et al. Declipping of audio signals using perceptual compressed sensing. *IEEE Transactions on Audio, Speech, and Language Processing*, 2013, 21.12: 2627-2637.
- [14] SIEDENBURG, Kai; KOWALSKI, Matthieu; DORFLER, Monika. Audio declipping with social sparsity. *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE*, 2014. p. 1577-1581.
- [15] KITIĆ, Srđan; BERTIN, Nancy; GRIBONVAL, Rémi. Audio declipping by cosparsity hard thresholding. *iTwist-2nd international-Traveling Workshop on Interactions between Sparse models and Technology*, 2014.
- [16] KITIĆ, Srđan; BERTIN, Nancy; GRIBONVAL, Rémi. Sparsity and cosparsity for audio declipping: a flexible non-convex approach. *International Conference on Latent Variable Analysis and Signal Separation. Springer, Cham*, 2015. p. 243-250.
- [17] ZÁVIŠKA, Pavel, et al. A proper version of synthesis-based sparse audio declipper. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019. p. 591-595.
- [18] JANSSEN, A. J. E. M.; VELDHUIS, R.; VRIES, L. Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1986, 34.2: 317-330.
- [19] HARVILLA, Mark J.; STERN, Richard M. Least squares signal declipping for robust speech recognition. *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
- [20] JONSCHER, Markus; SEILER, Juergen; KAUP, Andre. Declipping of Speech Signals Using Frequency Selective Extrapolation. *Speech Communication; 11. ITG Symposium; Proceedings of. VDE*, 2014. p. 1-4.
- [21] BILEN, Cagdas; OZEROV, Alexey; PÉREZ, Patrick. Audio declipping via nonnegative matrix factorization. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2015.
- [22] RAJMIC, Pavel. Řídké a nízkohodnostní reprezentace signálů s aplikacemi. *VUTIAM*, 2015.
- [23] SIEDENBURG, Kai; DÖRFLER, Monika. Structured sparsity for audio signals. *Proc. Int. Conf. on Digital Audio Effects (DAFx)*, 2011.
- [24] NAM, Sangnam, et al. The cosparsity analysis model and algorithms. *Applied and Computational Harmonic Analysis*, 2013, 34.1: 30-56.
- [25] SMÉKAL, Zdeněk. Analýza signálů a soustav: BASS. 2012.

- [26] CROCHIERE, R. A weighted overlap-add method of short-time Fourier analysis/synthesis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1980, 28.1: 99-102.
- [27] NARASIMHA, Madihally J. Modified overlap-add and overlap-save convolution algorithms for real signals. *IEEE Signal Processing Letters*, 2006, 13.11: 669-671.
- [28] In Wikipedia, The Free Encyclopedia: Overlap-add method. Retrieved 11:00, December 9, 2018, from <[https://en.wikipedia.org/w/index.php?title=Overlap%E2%80%93add\\_method&oldid=869397959](https://en.wikipedia.org/w/index.php?title=Overlap%E2%80%93add_method&oldid=869397959)>
- [29] GAULTIER, Clément; BERTIN, Nancy; GRIBONVAL, Rémi. CASCADE: Channel-Aware Structured Cospase Audio DEclipper. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018. p. 571-575.
- [30] PRŮŠA, Zdeněk, et al. LTFAT: A Matlab/Octave toolbox for sound processing. *Proc. 10th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2013. p. 299-314.
- [31] DOLBY LABORATORIES. Dolby Atmos [online]. Dostupné z: <<https://www.dolby.com/us/en/guide/dolby-atmos-trailers.html>>
- [32] SCHIMMEL, Jiří. Akustika a zvukové systémy. 2018.
- [33] Wikipedia contributors. MUSHRA. *Wikipedia, The Free Encyclopedia*. April 28, 2019. Dostupné z: <<https://en.wikipedia.org/w/index.php?title=MUSHRA&oldid=894530648>>.
- [34] ZÁVIŠKA, Pavel, et al. Revisiting synthesis model in sparse audio declipper. In: *International Conference on Latent Variable Analysis and Signal Separation*. Springer, Cham, 2018. p. 429-445.

## Seznam symbolů, veličin a zkratk

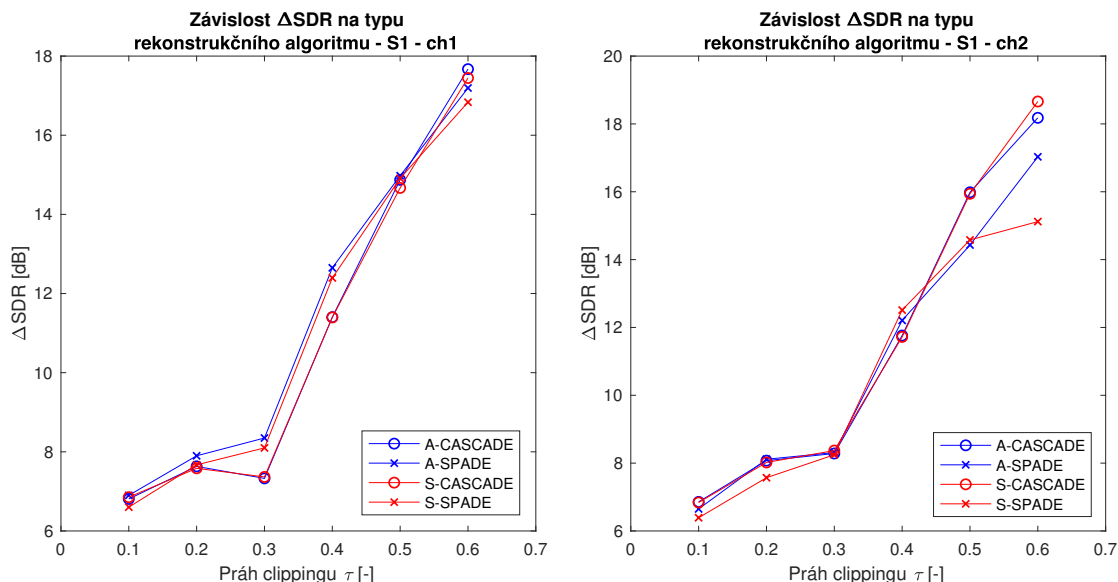
<b>A/D</b>	Analog-to-Digital – převod analogového signálu na digitální
<b>ADMM</b>	Alternating Direction Method of Multipliers – optimalizační algoritmus
<b>CASCADE</b>	Channel-Aware Structured Cospase Audio Declipper – rekonstrukční algoritmus saturovaných signálů
<b>cBAR</b>	Constrained Blind Amplitude Reconstruction – rekonstrukční algoritmus saturovaných signálů
<b>DAW</b>	Digital Audio Workstation – software pro záznam a editaci audio signálů
<b>FSE</b>	Frequency Selective Extrapolation – rekonstrukční algoritmus saturovaných signálů
<b>GEW</b>	Group Empirical Wiener – shrinkage operátor
<b>IHT</b>	Iterative Hard Thresholding – rekonstrukční algoritmus saturovaných singálů
<b>IHT-DC</b>	Iterative Hard Thresholding Declipping Constraints – rekonstrukční algoritmus saturovaných singálů
<b>LTFAT</b>	The Large Time/Frequency Analysis Toolbox – toolbox programu Matlab
<b>MUSHRA</b>	Multi Stimulus test with Hidden Reference and Anchor – metoda subjektivního poslechového testu
<b>NMF</b>	Non-negative Matrix Factorization – výpočetní postup
<b>ODG</b>	Objective Difference Grade – hodnotící škála metody PEAQ
<b>OLA</b>	Overlap-add – metoda přičítání přesahu při zpracování audio signálu po blocích
<b>OMP</b>	Orthogonal Matching Pursuit – rekonstrukční algoritmus saturovaných signálů
<b>PCSL1</b>	Perceptual Compressed Sensing with $\ell_1$ minimization – rekonstrukční algoritmus saturovaných signálů
<b>PEAQ</b>	Perceptual Evaluation of Audio Quality – ukazatel objektivně hodnotící kvalitu audio signálů podle subjektivního lidského vnímání
<b>PEW</b>	Persistent Empirical Wiener – shrinkage operátor
<b>RBAR</b>	Regularized Blind Amplitude reconstruction – rekonstrukční algoritmus saturovaných signálů
<b><math>R\ell_1</math>CC</b>	Reweighted $\ell_1$ minimization with clipping constraints – rekonstrukční algoritmus saturovaných singálů
<b>SDR</b>	Signal-to-Distortion Ratio – odstup signálu od zkreslení
<b>SNR</b>	Signal-to-Noise Ratio – odstup signálu od hluku
<b>SPADE</b>	Sparsity Audio Declipper – rekonstrukční algoritmus saturovaných signálů
<b>TPCC</b>	Trivial Pursuit With Clipping Constraints – rekonstrukční algoritmus saturovaných singálů
<b>VP</b>	Vektorový prostor
<b>WER</b>	Word Error Rate – objektivní ukazatel kvality signálu

# Seznam příloh

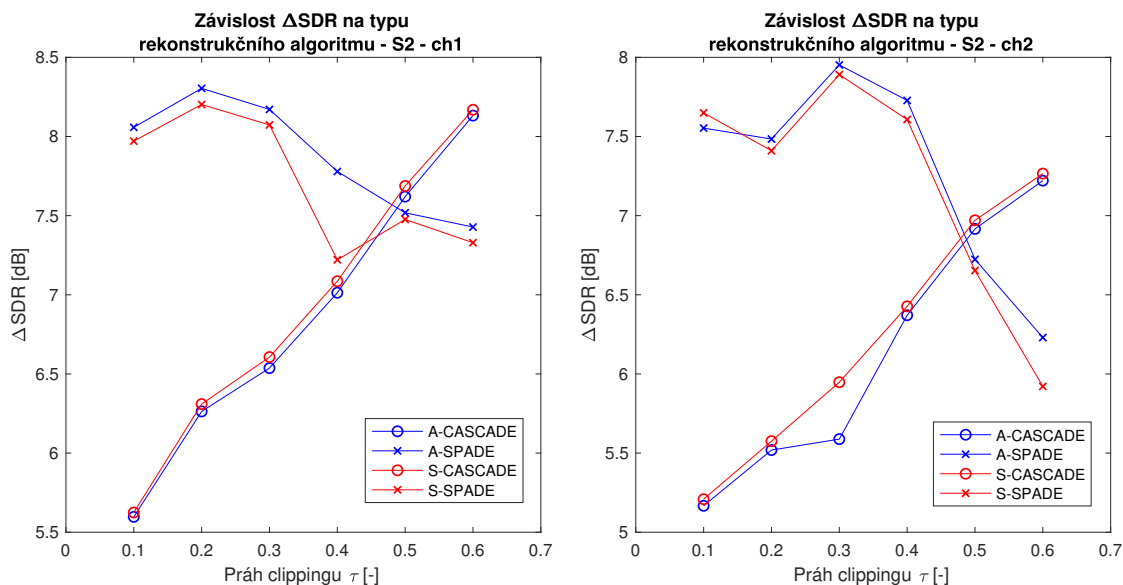
<b>A</b>	<b>Podrobné hodnocení výsledků rekonstrukčních algoritmů</b>	<b>62</b>
A.1	Ukazatel SDR . . . . .	62
A.2	Metoda PEAQ . . . . .	68
A.3	Subjektivní poslechový test . . . . .	74
<b>B</b>	<b>Obsah přiloženého CD</b>	<b>77</b>

# A Podrobné hodnocení výsledků rekonstrukčních algoritmů

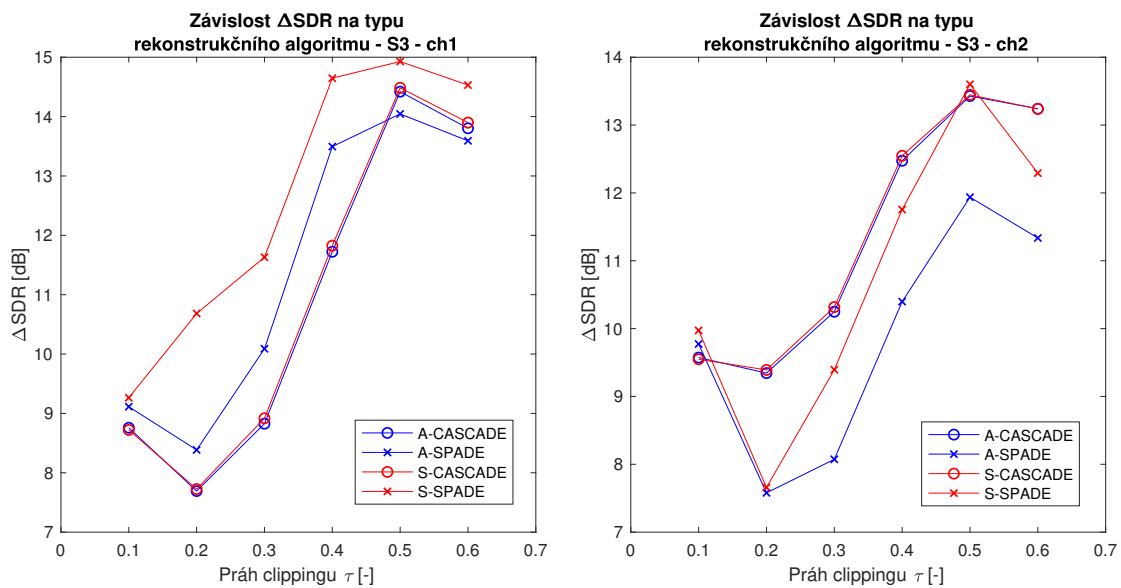
## A.1 Ukazatel SDR



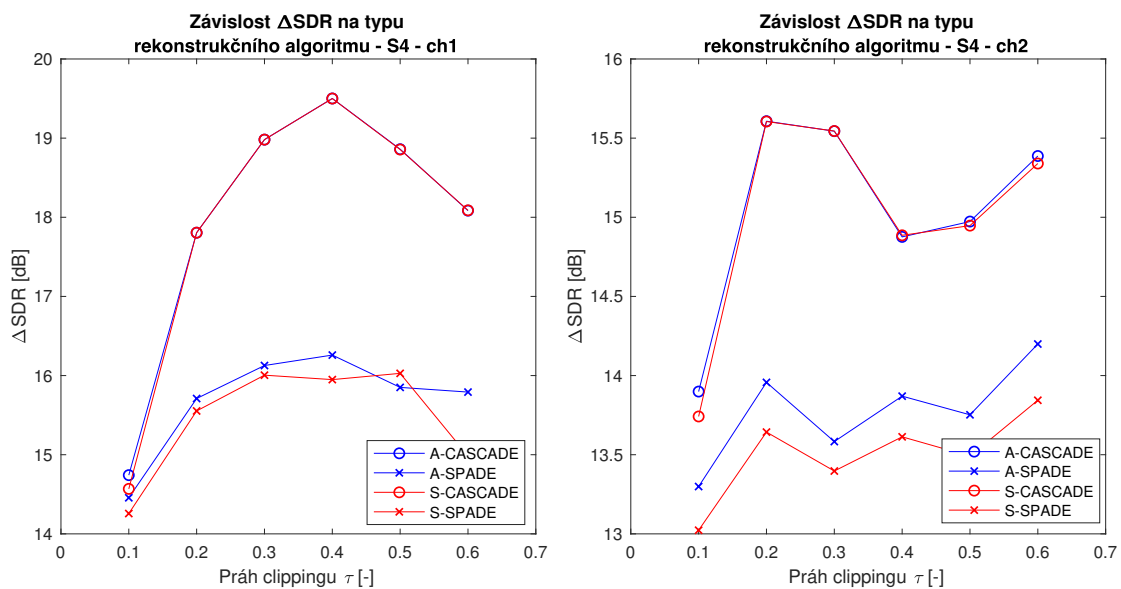
Obr. A.1: Závislost kvality rekonstrukce signálu S1 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta$ SDR.



Obr. A.2: Závislost kvality rekonstrukce signálu S2 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta$ SDR.

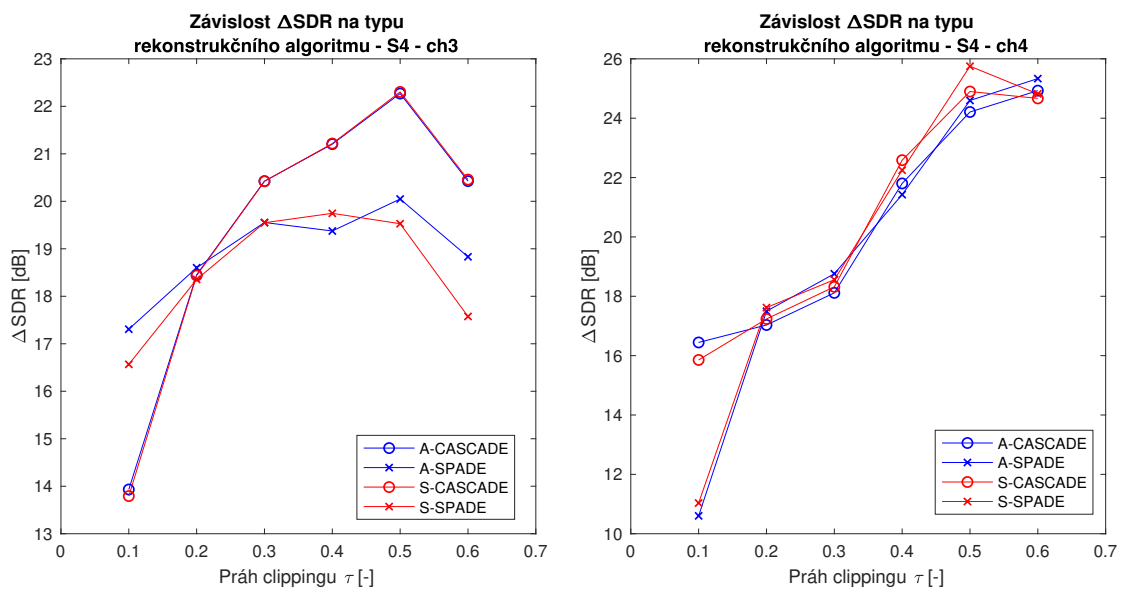


Obr. A.3: Závislost kvality rekonstrukce signálu S3 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta\text{SDR}$ .

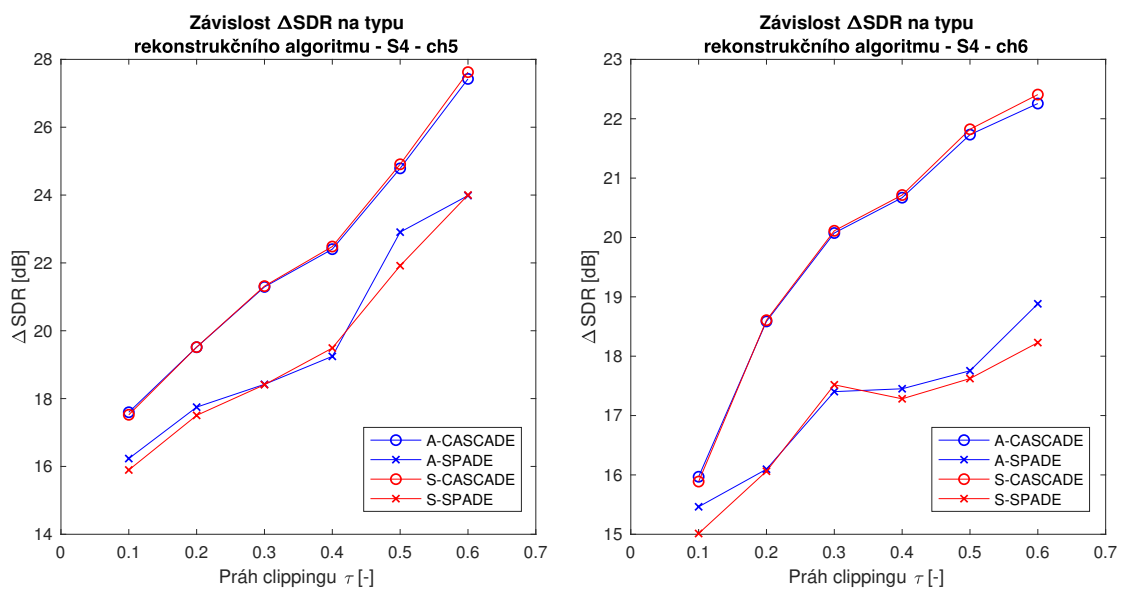


Obr. A.4: Závislost kvality rekonstrukce signálu S4 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta\text{SDR}$  (část 1).

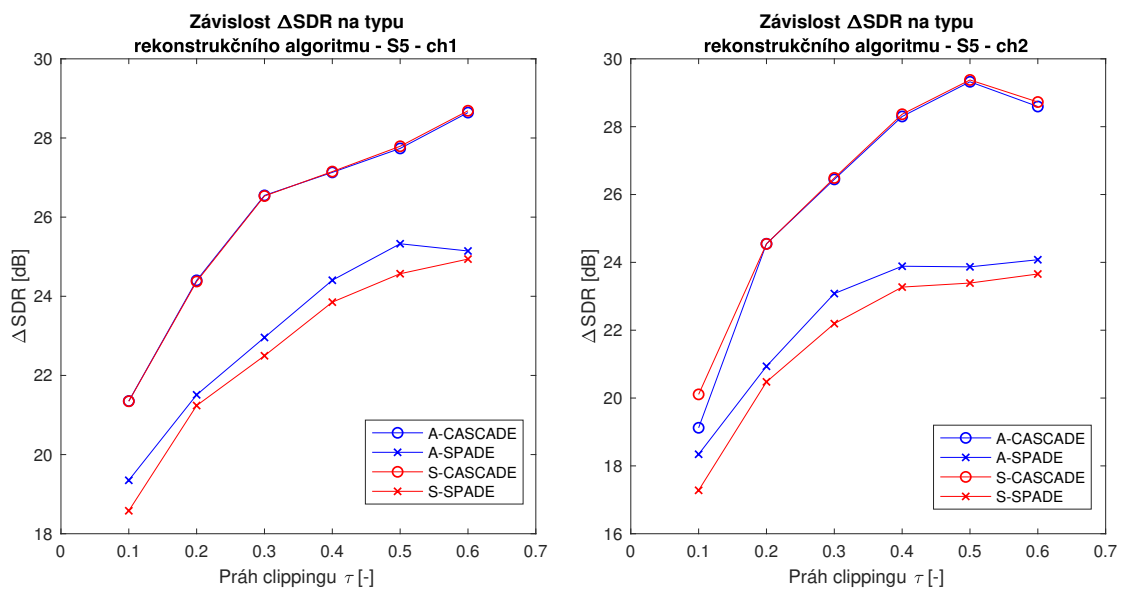




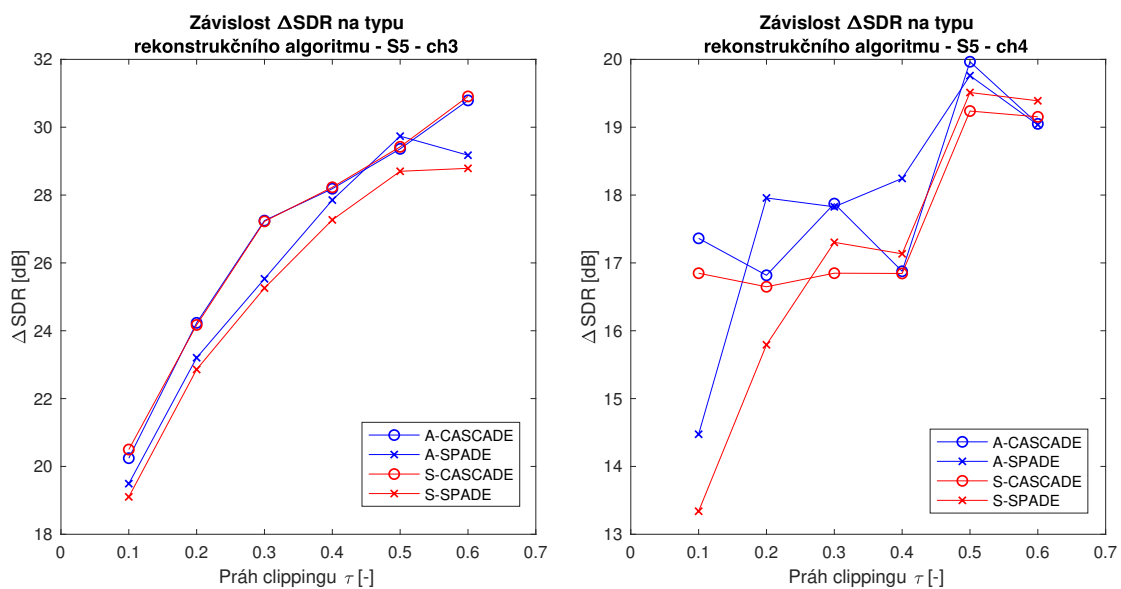
Obr. A.5: Závislost kvality rekonstrukce signálu S4 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta\text{SDR}$  (část 2).



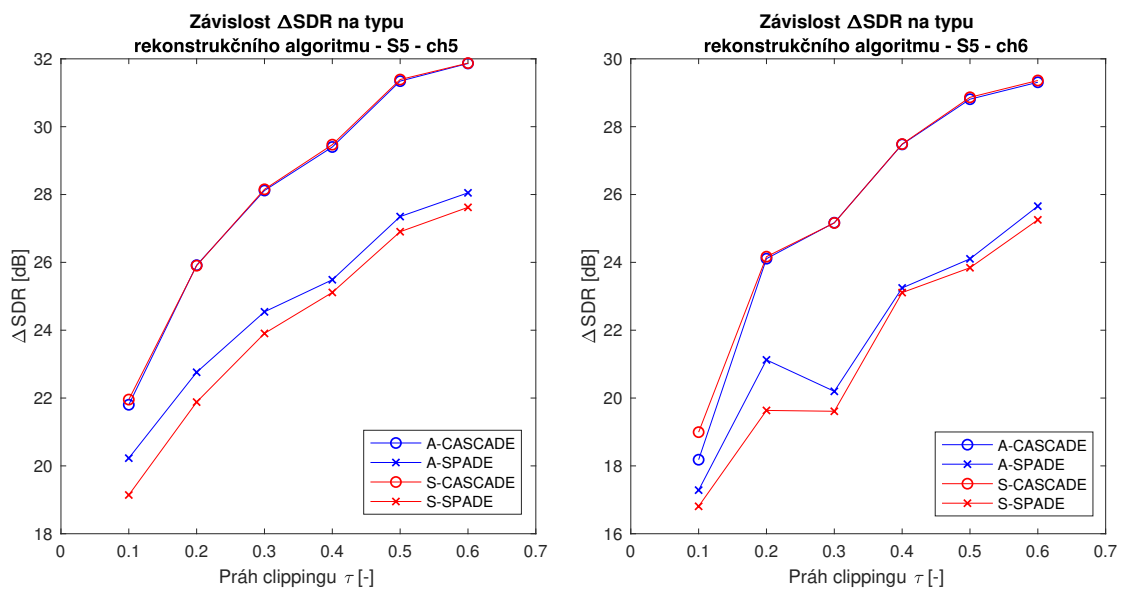
Obr. A.6: Závislost kvality rekonstrukce signálu S4 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta\text{SDR}$  (část 3).



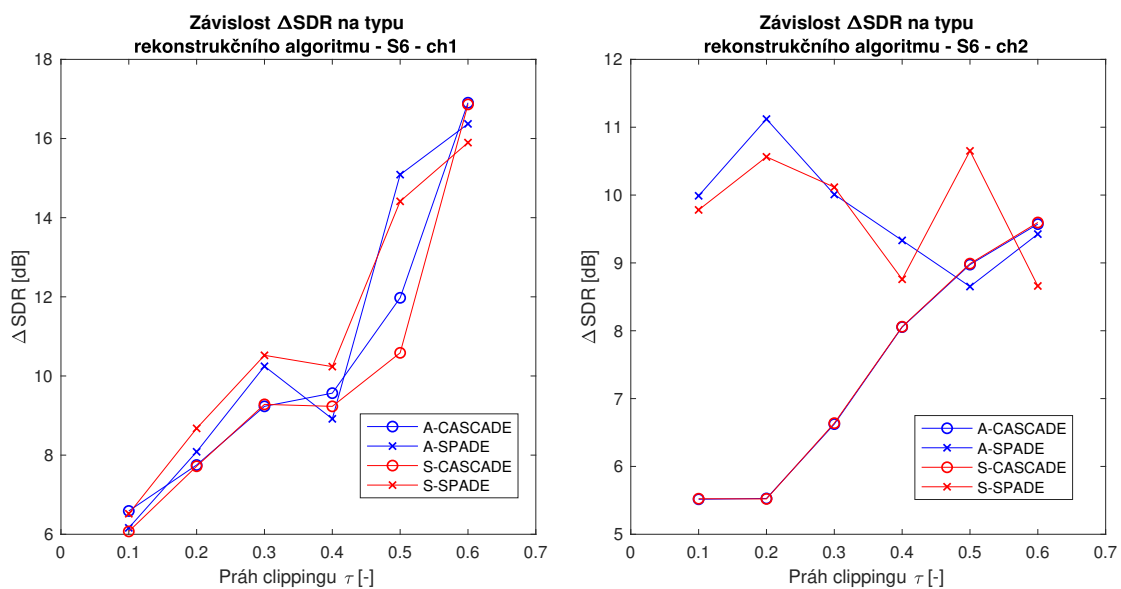
Obr. A.7: Závislost kvality rekonstrukce signálu S5 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta\text{SDR}$  (část 1).



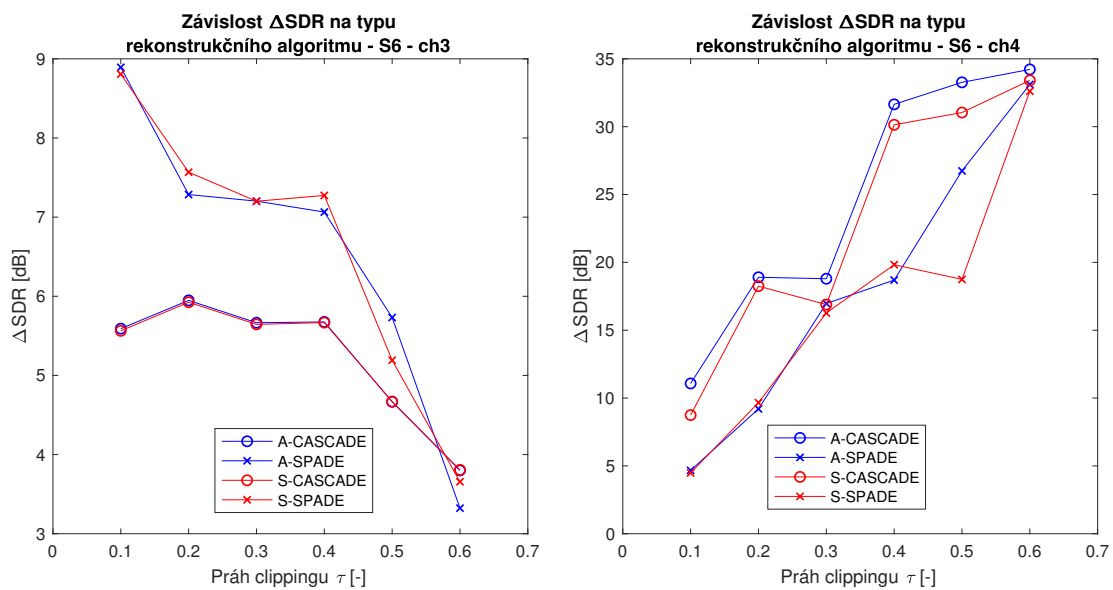
Obr. A.8: Závislost kvality rekonstrukce signálu S5 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta\text{SDR}$  (část 2).



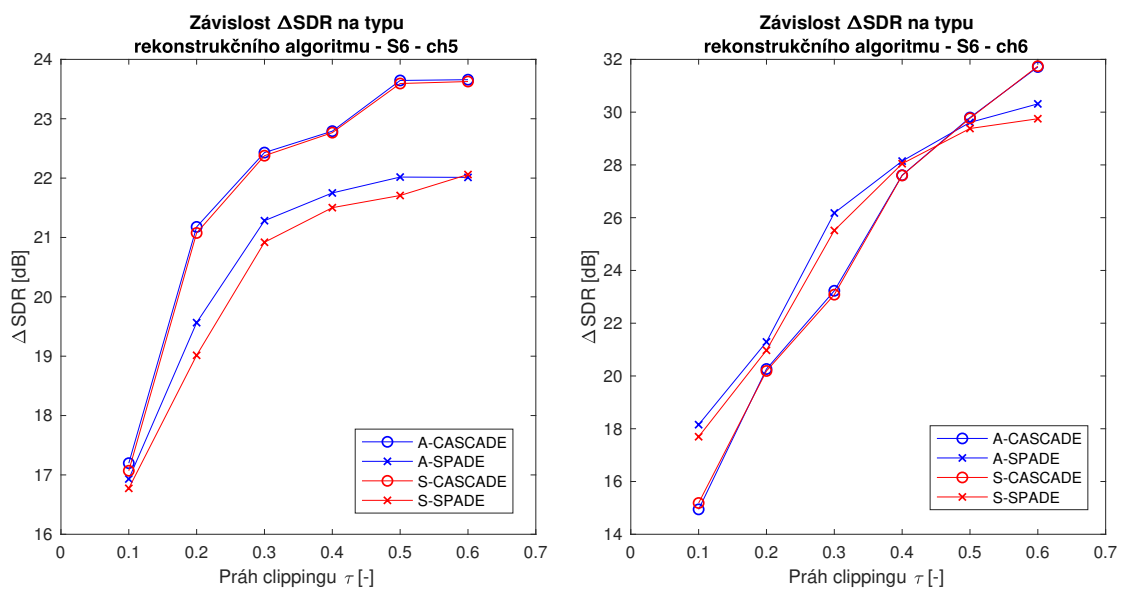
Obr. A.9: Závislost kvality rekonstrukce signálu S5 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta\text{SDR}$  (část 3).



Obr. A.10: Závislost kvality rekonstrukce signálu S6 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta\text{SDR}$  (část 1).

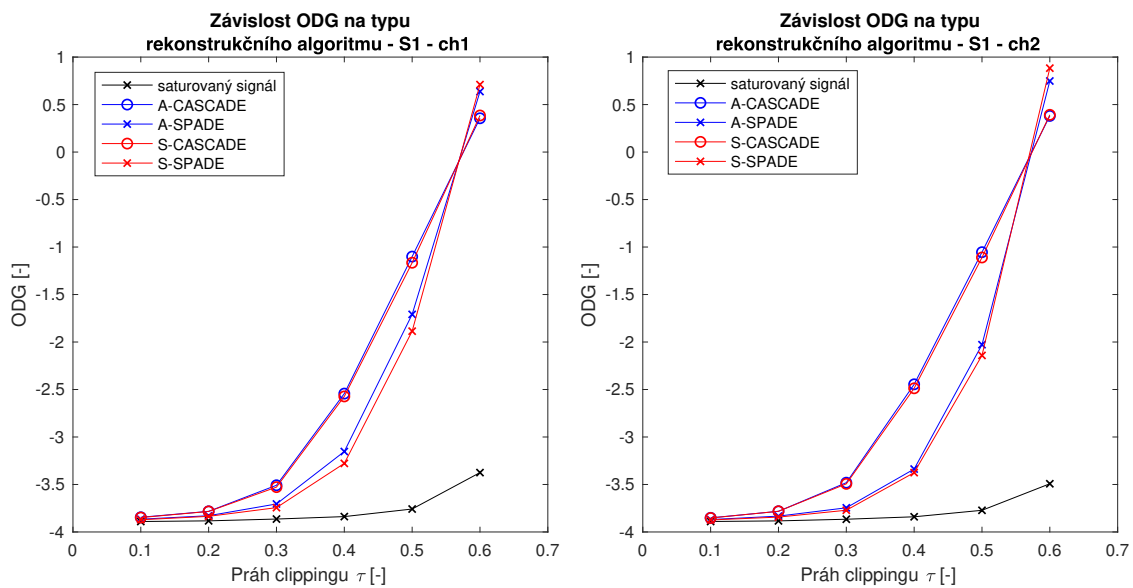


Obr. A.11: Závislost kvality rekonstrukce signálu S6 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta\text{SDR}$  (část 2).

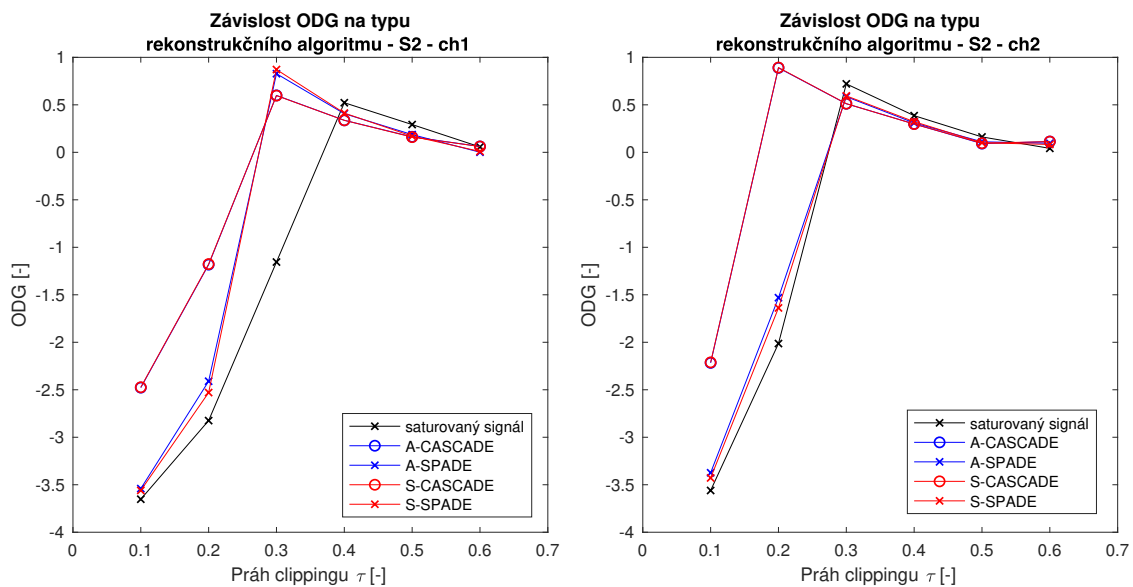


Obr. A.12: Závislost kvality rekonstrukce signálu S6 na typu rekonstrukčního algoritmu podle ukazatele  $\Delta\text{SDR}$  (část 3).

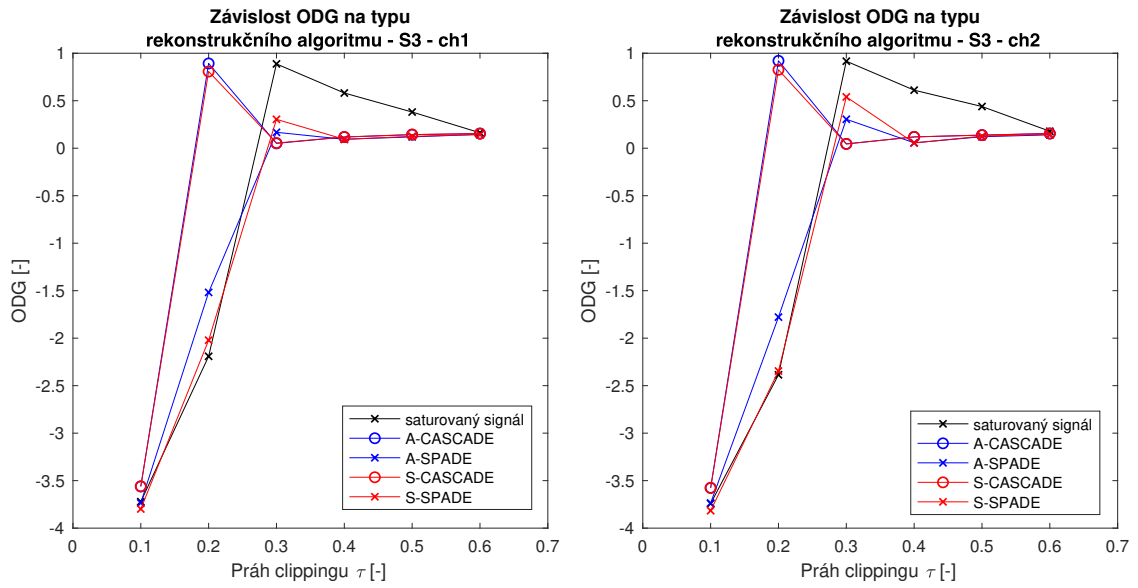
## A.2 Metoda PEAQ



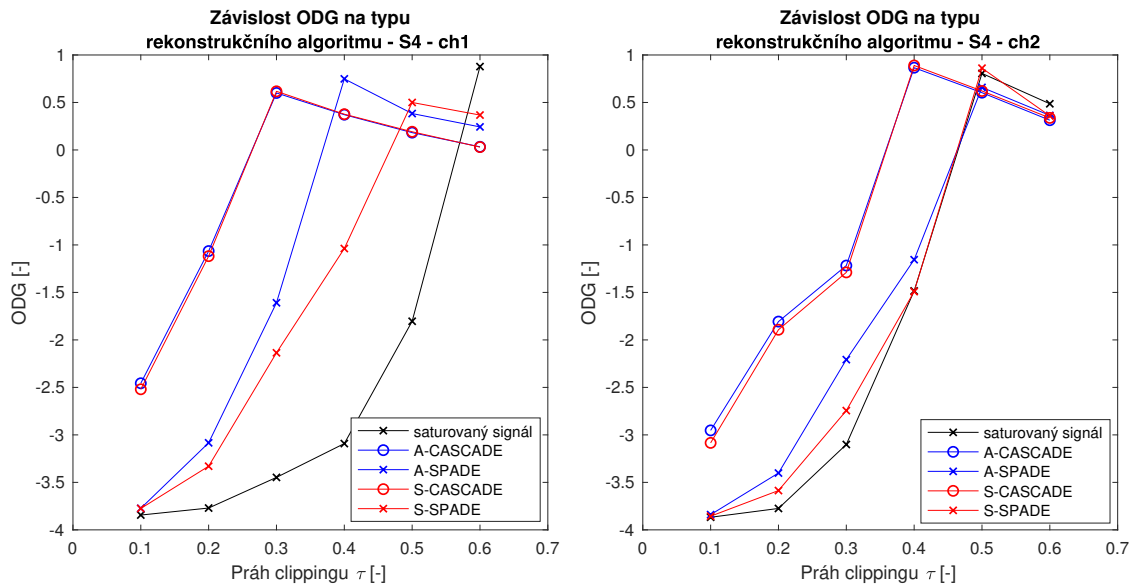
Obr. A.13: Závislost kvality rekonstrukce signálu S1 na typu rekonstrukčního algoritmu podle ukazatele PEAQ.



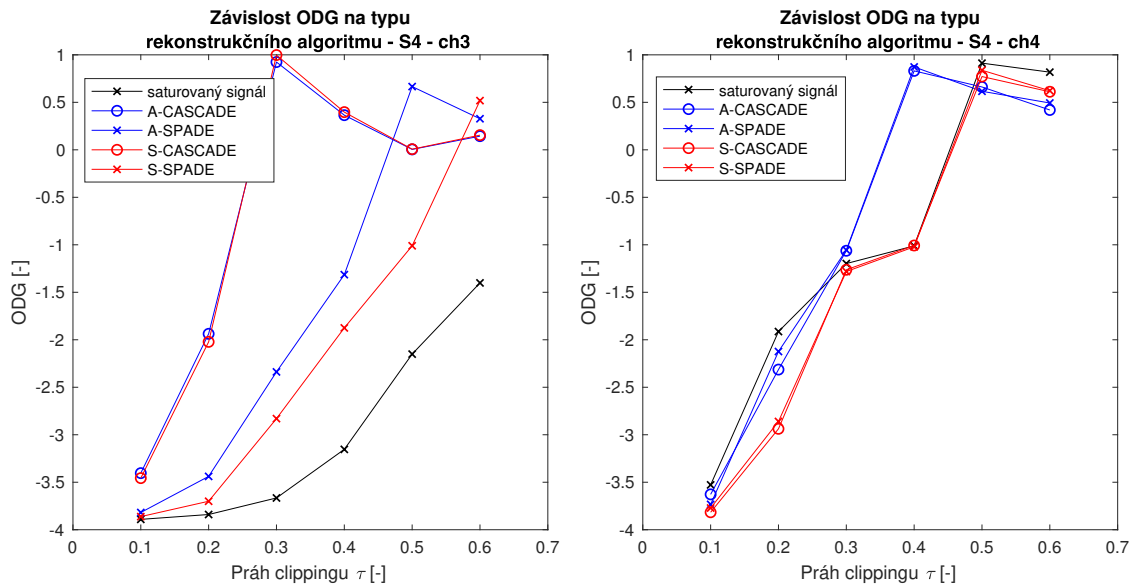
Obr. A.14: Závislost kvality rekonstrukce signálu S2 na typu rekonstrukčního algoritmu podle ukazatele PEAQ.



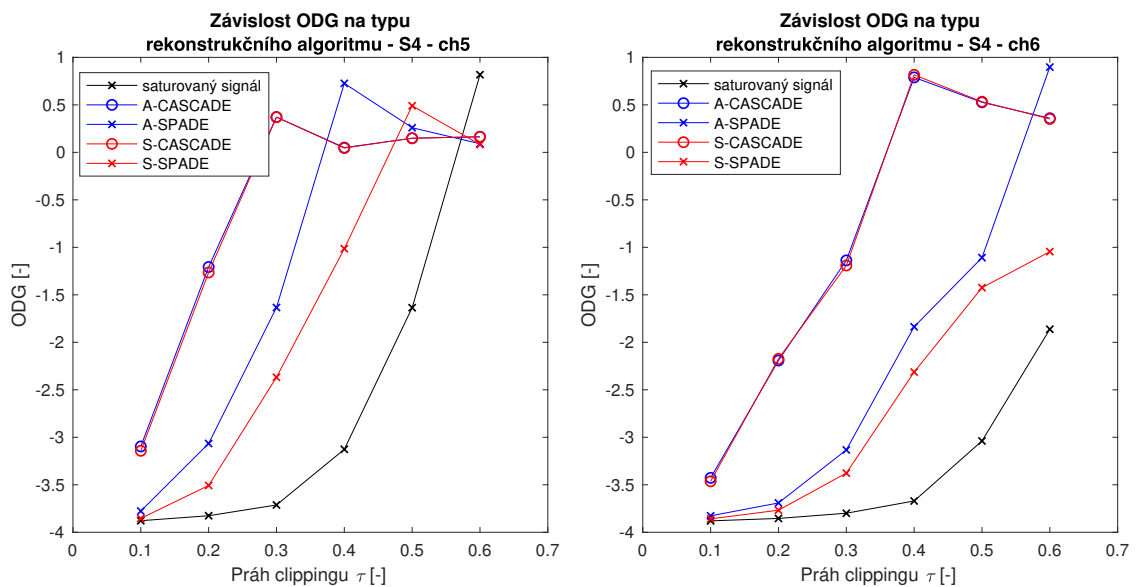
Obr. A.15: Závislost kvality rekonstrukce signálu S3 na typu rekonstrukčního algoritmu podle ukazatele PEAQ.



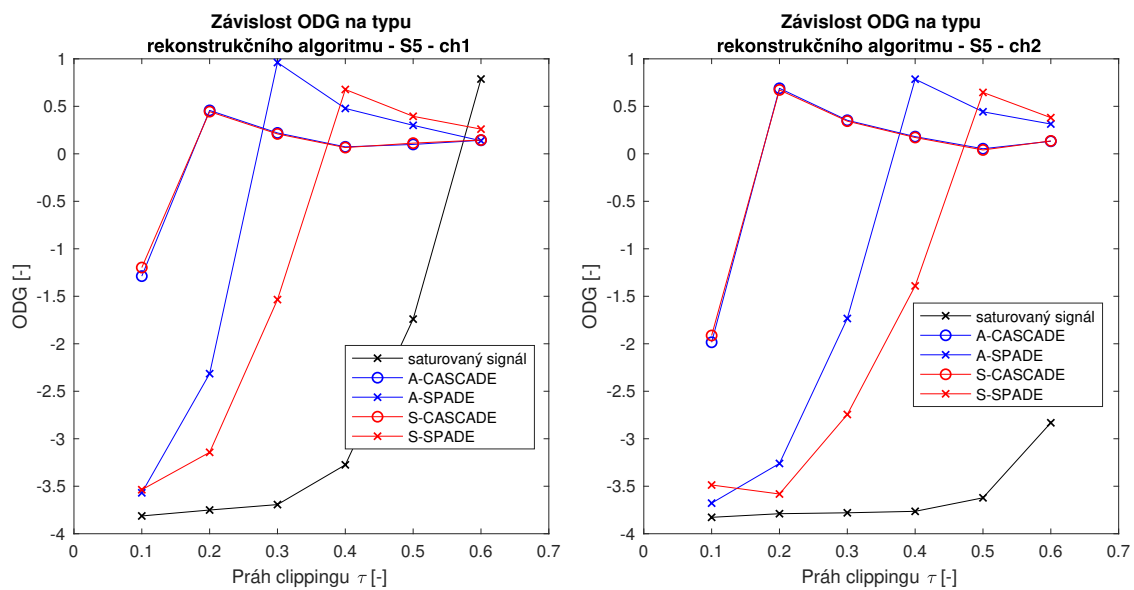
Obr. A.16: Závislost kvality rekonstrukce signálu S4 na typu rekonstrukčního algoritmu podle ukazatele PEAQ (část 1).



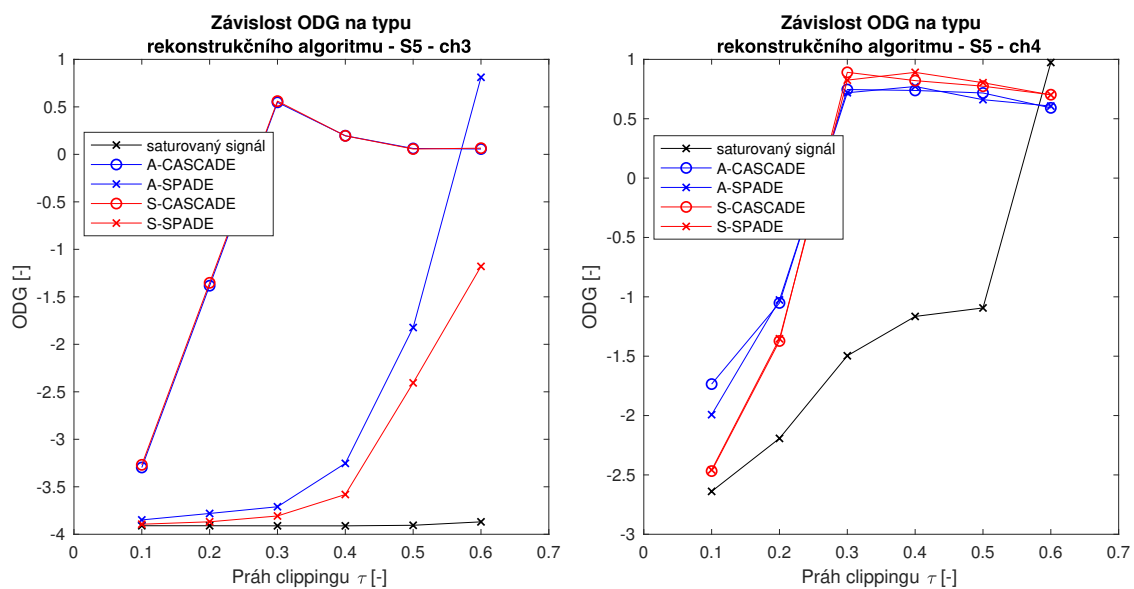
Obr. A.17: Závislost kvality rekonstrukce signálu S4 na typu rekonstrukčního algoritmu podle ukazatele PEAQ (část 2).



Obr. A.18: Závislost kvality rekonstrukce signálu S4 na typu rekonstrukčního algoritmu podle ukazatele PEAQ (část 3).

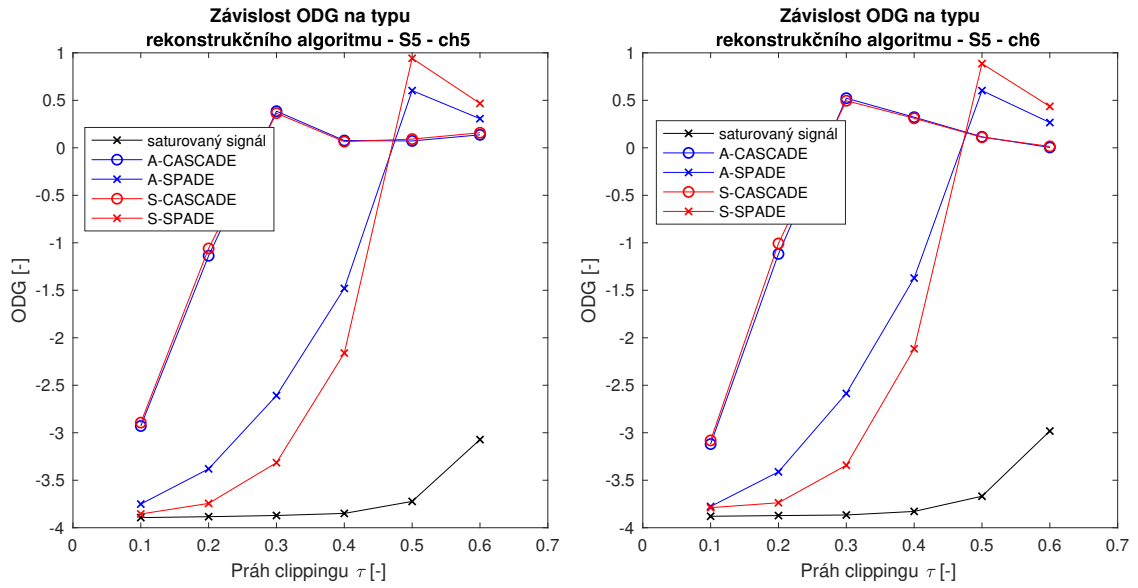


Obr. A.19: Závislost kvality rekonstrukce signálu S5 na typu rekonstrukčního algoritmu podle ukazatele PEAQ (část 1).

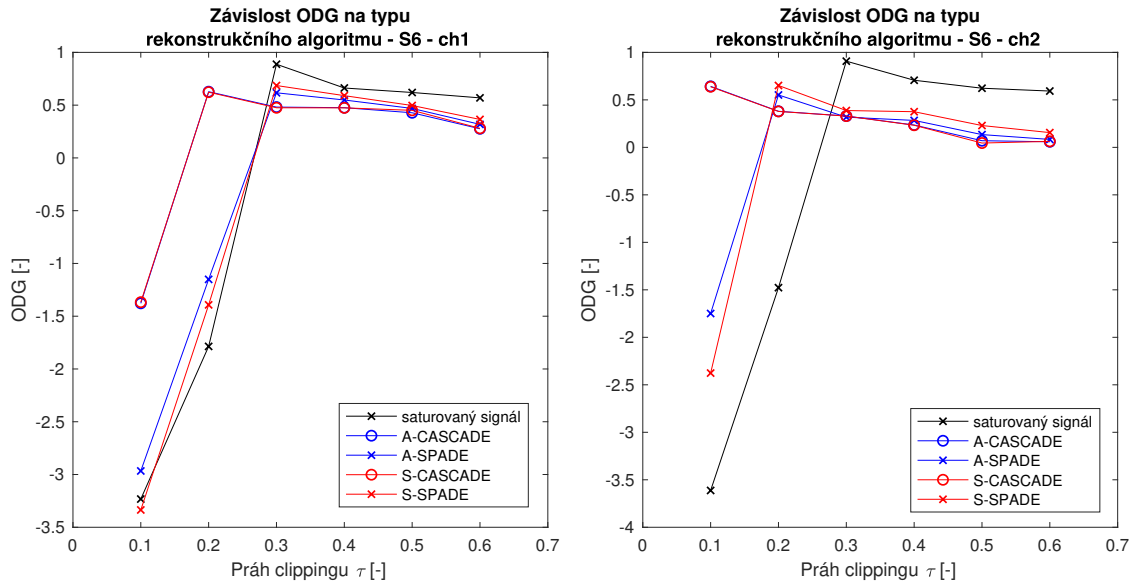


Obr. A.20: Závislost kvality rekonstrukce signálu S5 na typu rekonstrukčního algoritmu podle ukazatele PEAQ (část 2).

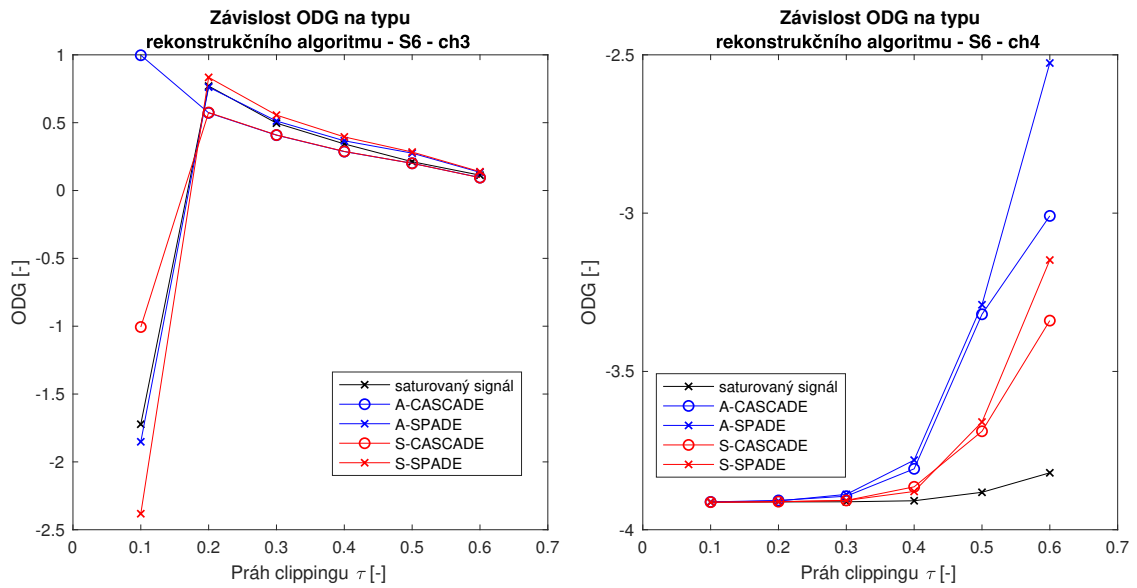




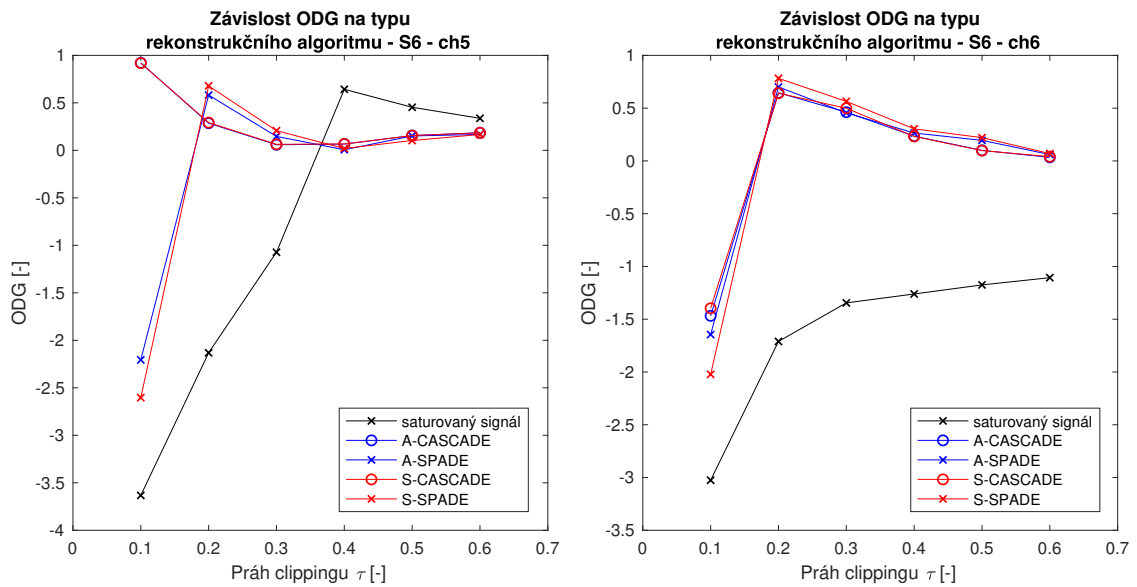
Obr. A.21: Závislost kvality rekonstrukce signálu S5 na typu rekonstrukčního algoritmu podle ukazatele PEAQ (část 3).



Obr. A.22: Závislost kvality rekonstrukce signálu S6 na typu rekonstrukčního algoritmu podle ukazatele PEAQ (část 1).



Obr. A.23: Závislost kvality rekonstrukce signálu S6 na typu rekonstrukčního algoritmu podle ukazatele PEAQ (část 2).



Obr. A.24: Závislost kvality rekonstrukce signálu S6 na typu rekonstrukčního algoritmu podle ukazatele PEAQ (část 3).

### A.3 Subjektivní poslechový test

Tab. A.1: Výsledky subjektivního poslechového testu - část 1 ze 3.

Ukázka	Práh clippingu $\tau$	Rekonstrukční algoritmus	Průměrné hodnocení	Interval
S2 - ch 2	0,1	Původní nahrávka	83,0	[76,9 ; 89,1]
S2 - ch 2	0,1	Saturovaná nahrávka	10,5	[4,2 ; 16,8]
S2 - ch 2	0,1	A-CASCADE	64,0	[54,4 ; 73,5]
S2 - ch 2	0,1	S-CASCADE	52,8	[40,8 ; 64,8]
S2 - ch 2	0,1	A-SPADE	54,3	[42,4 ; 66,1]
S2 - ch 2	0,1	S-SPADE	60,5	[49,7 ; 71,3]

S2 - ch 2	0,3	Původní nahrávka	72,2	[63,1 ; 81,3]
S2 - ch 2	0,3	Saturovaná nahrávka	54,9	[43,5 ; 66,3]
S2 - ch 2	0,3	A-CASCADE	69,8	[59,3 ; 80,2]
S2 - ch 2	0,3	S-CASCADE	75,0	[64,8 ; 85,2]
S2 - ch 2	0,3	A-SPADE	75,4	[65,9 ; 84,8]
S2 - ch 2	0,3	S-SPADE	71,4	[60,6 ; 82,1]

S2 - ch 2	0,5	Původní nahrávka	77,1	[70,2 ; 83,9]
S2 - ch 2	0,5	Saturovaná nahrávka	46,0	[36,4 ; 55,6]
S2 - ch 2	0,5	A-CASCADE	75,6	[64,6 ; 86,6]
S2 - ch 2	0,5	S-CASCADE	70,8	[61,3 ; 80,2]
S2 - ch 2	0,5	A-SPADE	77,8	[69,3 ; 86,2]
S2 - ch 2	0,5	S-SPADE	71,1	[60,4 ; 81,7]

S3 - ch1	0,1	Původní nahrávka	84,0	[77,3 ; 90,7]
S3 - ch1	0,1	Saturovaná nahrávka	17,8	[11,9 ; 23,6]
S3 - ch1	0,1	A-CASCADE	61,6	[52,3 ; 70,9]
S3 - ch1	0,1	S-CASCADE	61,0	[52,7 ; 69,3]
S3 - ch1	0,1	A-SPADE	46,3	[36,3 ; 56,2]
S3 - ch1	0,1	S-SPADE	39,8	[30,8 ; 48,7]

S3 - ch1	0,3	Původní nahrávka	78,1	[70,0 ; 86,2]
S3 - ch1	0,3	Saturovaná nahrávka	75,1	[66,7 ; 83,4]
S3 - ch1	0,3	A-CASCADE	79,0	[68,8 ; 89,1]
S3 - ch1	0,3	S-CASCADE	80,4	[72,8 ; 87,9]
S3 - ch1	0,3	A-SPADE	71,7	[62,9 ; 80,5]
S3 - ch1	0,3	S-SPADE	70,0	[60,1 ; 79,8]

Tab. A.2: Výsledky subjektivního poslechového testu - část 2 ze 3.

Ukázka	Práh clippingu $\tau$	Rekonstrukční algoritmus	Průměrné hodnocení	Interval
S3 - ch1	0,5	Původní nahrávka	78,4	[69,7 ; 87,0]
S3 - ch1	0,5	Saturovaná nahrávka	78,5	[69,5 ; 87,5]
S3 - ch1	0,5	A-CASCADE	78,3	[70,2 ; 86,4]
S3 - ch1	0,5	S-CASCADE	81,7	[74,0 ; 89,4]
S3 - ch1	0,5	A-SPADE	81,7	[73,7 ; 89,7]
S3 - ch1	0,5	S-SPADE	77,3	[68,8 ; 85,8]

S5 - ch5	0,1	Původní nahrávka	86,3	[81,3 ; 91,2]
S5 - ch5	0,1	Saturovaná nahrávka	7,5	[3,8 ; 11,2]
S5 - ch5	0,1	A-CASCADE	80,5	[73,0 ; 88,0]
S5 - ch5	0,1	S-CASCADE	77,3	[70,6 ; 83,9]
S5 - ch5	0,1	A-SPADE	50,5	[41,5 ; 59,5]
S5 - ch5	0,1	S-SPADE	43,6	[35,7 ; 51,4]

S5 - ch5	0,3	Původní nahrávka	82,5	[76,6 ; 88,3]
S5 - ch5	0,3	Saturovaná nahrávka	11,3	[6,1 ; 16,4]
S5 - ch5	0,3	A-CASCADE	82,5	[74,7 ; 90,2]
S5 - ch5	0,3	S-CASCADE	77,3	[69,3 ; 85,2]
S5 - ch5	0,3	A-SPADE	72,8	[65,3 ; 80,2]
S5 - ch5	0,3	S-SPADE	58,0	[48,7 ; 67,3]

S5 - ch5	0,5	Původní nahrávka	82,0	[76,0 ; 88,0]
S5 - ch5	0,5	Saturovaná nahrávka	22,8	[17,0 ; 28,5]
S5 - ch5	0,5	A-CASCADE	78,4	[71,4 ; 85,4]
S5 - ch5	0,5	S-CASCADE	84,4	[79,3 ; 89,4]
S5 - ch5	0,5	A-SPADE	77,8	[71,0 ; 84,5]
S5 - ch5	0,5	S-SPADE	69,0	[59,6 ; 78,4]

S6 - ch1	0,1	Původní nahrávka	77,0	[70,6 ; 83,4]
S6 - ch1	0,1	Saturovaná nahrávka	11,3	[5,5 ; 17,0]
S6 - ch1	0,1	A-CASCADE	64,5	[53,3 ; 75,7]
S6 - ch1	0,1	A-CASCADE	61,3	[50,8 ; 71,7]
S6 - ch1	0,1	S-SPADE	60,6	[48,5 ; 72,7]
S6 - ch1	0,1	S-SPADE	38,3	[26,8 ; 49,7]

Tab. A.3: Výsledky subjektivního poslechového testu - část 3 ze 3.

Ukázka	Práh clippingu $\tau$	Rekonstrukční algoritmus	Průměrné hodnocení	Interval
S6 - ch1	0,3	Původní nahrávka	79,7	[73,1 ; 86,2]
S6 - ch1	0,3	Saturovaná nahrávka	49,0	[37,6 ; 60,4]
S6 - ch1	0,3	A-CASCADE	79,5	[73,2 ; 85,7]
S6 - ch1	0,3	A-CASCADE	75,3	[67,8 ; 82,7]
S6 - ch1	0,3	S-SPADE	79,3	[73,1 ; 85,4]
S6 - ch1	0,3	S-SPADE	73,3	[67,4 ; 79,1]

S6 - ch1	0,5	Původní nahrávka	81,0	[72,6 ; 89,4]
S6 - ch1	0,5	Saturovaná nahrávka	72,5	[64,0 ; 81,0]
S6 - ch1	0,5	A-CASCADE	82,8	[76,7 ; 88,8]
S6 - ch1	0,5	A-CASCADE	81,5	[74,2 ; 88,8]
S6 - ch1	0,5	S-SPADE	85,5	[80,2 ; 90,7]
S6 - ch1	0,5	S-SPADE	82,9	[76,9 ; 88,8]

## B Obsah přiloženého CD

diplomova_prace.pdf .....	text práce
algoritmus .....	složka obsahující kompletní rekonstrukční algoritmus
ACASCADE.m	
ASPADE.m	
clipping.m	
declipping.m	
hard_thresholding.m	
projection.m	
SCASCADE.m	
sdr_func.m	
segmentation.m	
SSPADE.m	
vysledky	
nastaveni.xlsx .....	průběžné výsledky z nastavování algoritmu CASCADE
SPADE-CASCADE.xlsx .....	konečné porovnání algoritmů SPADE a CASCADE
MUSHRA.xlsx .....	Kompletní výsledky subjektivního poslechového testu
vzorky .....	složka s testovacími nahrávkami
S1.wav	
S2.wav	
S3.wav	
S4.wav	
S5.wav	
S6.wav	